

AD-A244 420



②

J. W. Freeman
R. A. Wolf
R. W. Spiro
G-H. Voigt
B. A. Hausman

B. A. Bales
R. V. Hilmer
A. Nagai
R. Lambour

DTIC
ELECTE
DEC 05 1991

30 June 1991

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED



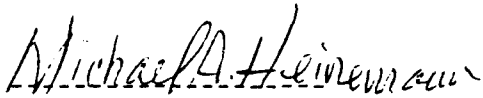
PHILLIPS LABORATORY
AIR FORCE SYSTEMS COMMAND
HANSCOM AIR FORCE BASE, MASSACHUSETTS 01731-5000

91-16856

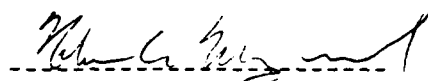


01 1 20 00 00

This technical report has been reviewed and is approved for publication.

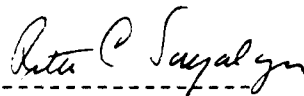


MICHAEL A. HEINEMANN
Contract Manager
Space Plasmas and Fields Br.
Space Physics Division



NELSON C. MAYNARD, Chief
Space Plasmas and Fields Br.
Space Physics Division

FOR THE COMMANDER



RITA C. SAGALYN
Director
Space Physics Division

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/TSI, Hanscom AFB, MA 01731. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION NUMBER		Form Approved OMB NO. 0754-0188	
Public report (this report is available to the public) Unannounced (this report is available to the public) Announced (this report is available to the public)		Public report (this report is available to the public) Unannounced (this report is available to the public) Announced (this report is available to the public)	
1. AGENCY USE ONLY (leave blank) 30 June 1991		7. REPORT TYPE AND DATES COVERED Final(15 Nov 1987-31 Mar 1990) Vol II	
4. TITLE AND SUBTITLE Development of a Magnetospheric Specification Model Volume II: Appendices A, B, C, D, E		5. FUNDING NUMBERS PE 62101F PR 2688 TA 04 WU GB	
6. AUTHOR(S) J. W. Freeman, R.A. Wolf, R. W. Spiro, G-H. Voigt, B. A. Hausman, B. A. Bales, R. V. Hilmer, A. Nagai, R. Lambour		Contract: F19628-87-K-0001	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Rice University Department of Space Physics & Astronomy Houston, TX 77251		8. PERFORMING ORGANIZATION REPORT NUMBER PL-TR-91-2276 (II)	
9. PHILLIPS LABORATORY Hanscom AFB, MA 01731-5000			
Contract Manager: Michael Heinemann/PHG			
11. SUPPLEMENTARY NOTES			
12. DISTRIBUTION STATEMENT (if applicable) Approved for public release; Distribution unlimited		13. DISTRIBUTION CODE	
<p>Rice University has developed a practical computer model that is capable of specifying electron and ion fluxes in the middle magnetosphere during geomagnetic storms. The model, called the Magnetospheric Specification Model (MSM), uses ground-based and satellite data from the Space Forecast Center-Environment Data Base to establish initial and boundary conditions and to determine input parameters for the magnetic and electric field models. These input values are updated every 15 minutes, and new output fluxes are computed for the same times. The primary function of the MSM is the specification of fluxes of 1-100 KeV electrons in the geostationary orbit region. However, it is also designed to specify a broad range of additional parameters for the global ionospheric-magnetospheric system, including fluxes of 1-50 KeV ions, auroral electron precipitation and ionospheric electric fields.</p>			
14. SUBJECT TERMS	Spacecraft charging	Magnetic storms	414
Electron fluxes	Radiation belts	Proton fluxes	
Ion fluxes	Space radiation	Geomagnetic storms	
Magnetosphere	Trapped radiation	Energetic particles	
15. SECURITY CLASSIFICATION	16. SECURITY CLASSIFICATION	17. SECURITY CLASSIFICATION	18. SECURITY CLASSIFICATION
Unclassified	Unclassified	Unclassified	SAR

CONT OF BLOCK 13:

The model is accompanied by an application program that allows specification of fluxes at an arbitrary point in the magnetosphere within the modeling region. Consistent with its primary function, the MSM has been tested against spacecraft data for 2 substantial storms and has been shown to produce a good characterization of the enhancements of 40 KeV electron fluxes in the equatorial plane. The model never failed to predict high fluxes when they were observed, although it did predict high fluxes in some cases when they were not observed and it did fail to predict flux dropouts observed by the spacecraft. The MSM is ready for adaptation for use in an operational setting where the goal is real-time and retrospective specification of hazardous charged particle fluxes associated with geomagnetic storms.

Accession For	
NTIS	CRA&I
DTIC	TAB
Unannounced	
Justification	
By	
Date	
Approved by	
DLI	
A-1	



Appendix A

FUNCTIONAL DESCRIPTION

FUNCTIONAL DESCRIPTION - 1

Functional Description

2.2 Objectives. The Magnetospheric Specification Model was developed as a set of algorithms to be developed into a real-time program for use in the Space Forecast Center operated by the Air Weather Service of the U.S. Air Force. It is intended to help the Center personnel perform their mission of providing information, particularly during magnetospheric disturbances, to "customers" who operate spacecraft. The output of the model includes (but is not limited to):

- 1) Fluxes of electrons and ions in the inner plasma sheet and the geosynchronous orbit region.
- 2) Energy fluxes and characteristic energies of electrons precipitated into the auroral ionosphere.

2.4 Proposed Methods and Procedures.

The most sophisticated type of theoretical inner-magnetosphere/ionosphere model is exemplified by the Rice Convection Model (RCM). However, the Air Force requirement for near-real time data output and the relatively slow run-time of the RCM made it unsuitable for Air Force operational use. With these constraints in mind we dropped the self-consistent treatment and have separate ionospheric electric field and magnetospheric magnetic field models to trace particles in the magnetosphere equatorial plane (See attached flow chart). The magnetic field model provides the mapping between the two regions with the condition that the equatorward edge of the auroral zone in the ionosphere maps to the inner edge of the plasma sheet in the magnetospheric equatorial plane.

3.1 Specific Performance Requirements.

The output of the model includes fluxes of electrons and ions in the inner plasma sheet and the geosynchronous orbit region and energy fluxes and characteristic energies of the electrons and ions precipitated into the ionosphere. The throughput time will vary depending on the number and energy of particles traced, as well as the length of time for which the model is to be run.

3.2.1 Accuracy and Validity. The Magnetospheric Specification Model produces the most accurate and reliable specifications possible given the available input parameters and the state of magnetospheric science. The standard deviation of the model fluxes and available observed geosynchronous satellite data is output after each run.

3.2.2 Timing. Attached is a report on the throughput time on the IBM 3081 computer. All input data for the run will be acquired from the operational Environmental Database at the beginning of the run as shown on the MSM Control and Data Flow Diagram. All output data will be returned to the Environmental Database.

FUNCTIONAL DESCRIPTION - 2

3.2 Functional Area System Functions. The major components of the system are:

MSMCON	Main program - sets start, ending, and time steps for the run
PPTCON	Executive routine to control flow of Magnetospheric Specification Model
EBCON	Executive routine to control calculation of electric and magnetic field models
PPTM	Subroutine to trace particle trajectories from each grid point for each particle species back in time from time = TSTART to time = TSTOP

Attached is flow chart of each of the major components.

3.3 Inputs and Outputs.

Following is input within the program:

ISTART(1)	Start Year (last two digits)
ISTART(2)	Start Day (julian day). January 1 is day 1.
ISTART(3)	Start Time (seconds)
IEND(1)	End Year (last two digits)
IEND(2)	End Day (julian day) January 1 is day 1.
IEND(2)	End Time (seconds)
IINC(1)	Program time step increment (year)
IINC(2)	Program time step increment (day)
IINC(3)	Program time step increment (seconds)
IRECEB	For restart, record number of previously calculated electric and magnetic field output data to use as input for current run
IRECPT	For restart, record number of previously calculated particle-trace output data to use as input for current run

Following are the input files required, including those to be replace by calls to the environmental data base:

BOxxxxxx	The magnetic field matrices needed for the model run.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.

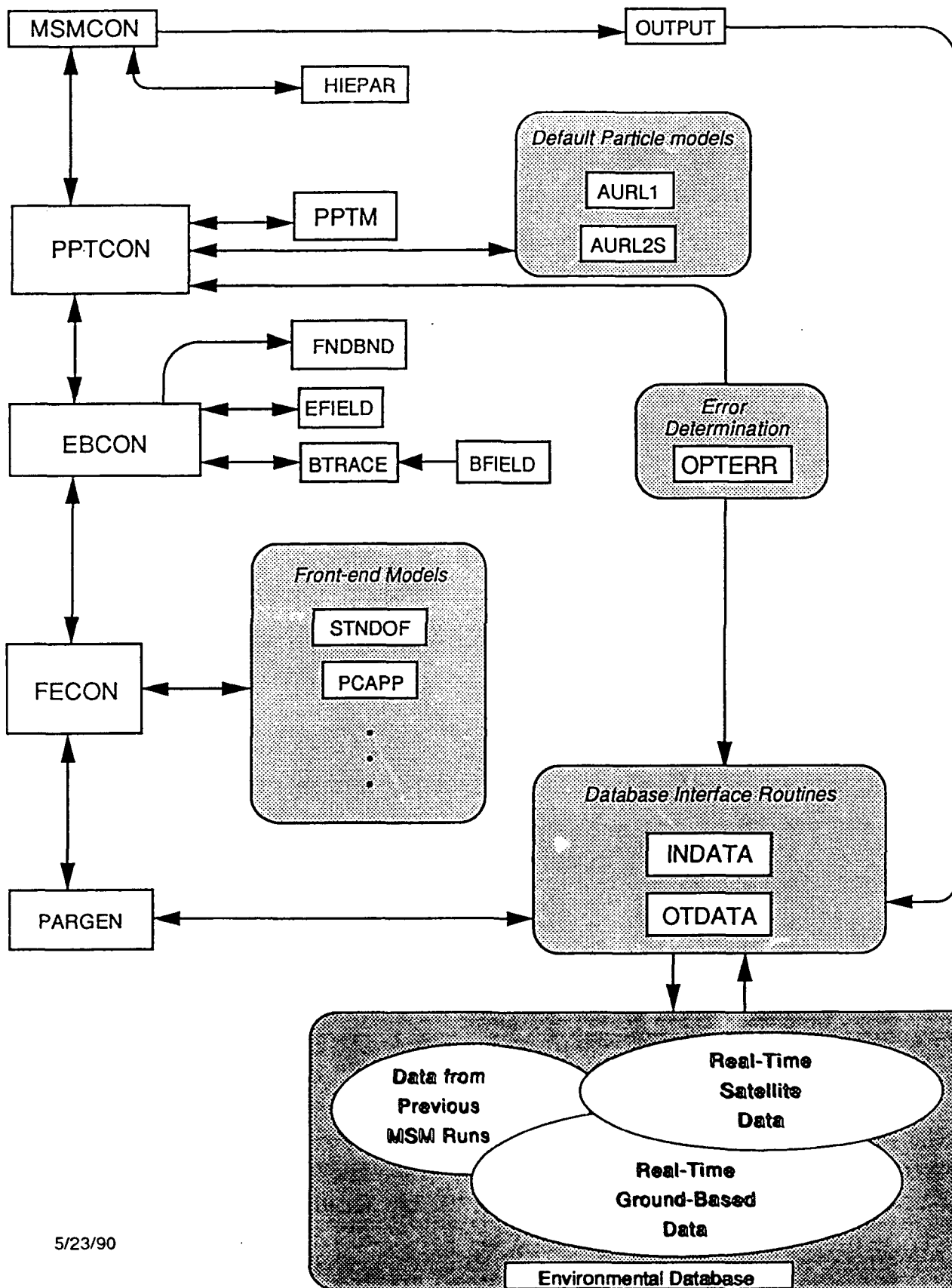
FUNCTIONAL DESCRIPTION - 3

ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm ³)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

Following are the output files:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts).
VNORTH	Northern hemisphere electric potential distribution (Volts).
VSOUTH	Southern hemisphere electric potential distribution (Volts).
VM	Flux tube volume ((Re/nT) ^{-2/3}).
BMIN	Equatorial magnetic field strength (nT).
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
BNDLOC	Location of outer boundary of detailed particle traces.
EFLUX	Precipitating energy flux array (ergs/cm ² -sec).
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec).
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV).
AUGPAR	Augmented data array for input values.
COLAT	Grid colatitude array (radians).
ALOCT	Grid local time array (radians eastward from noon).
FLUX	Flux values for all energy channels at all grid points

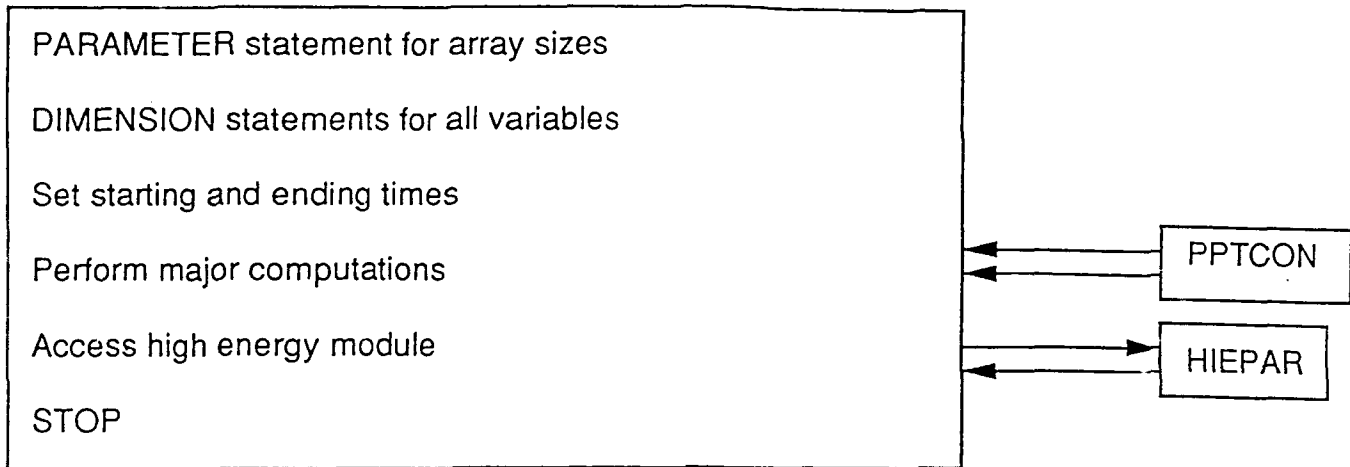
MSM Control & Data Flow Diagram



5/23/90

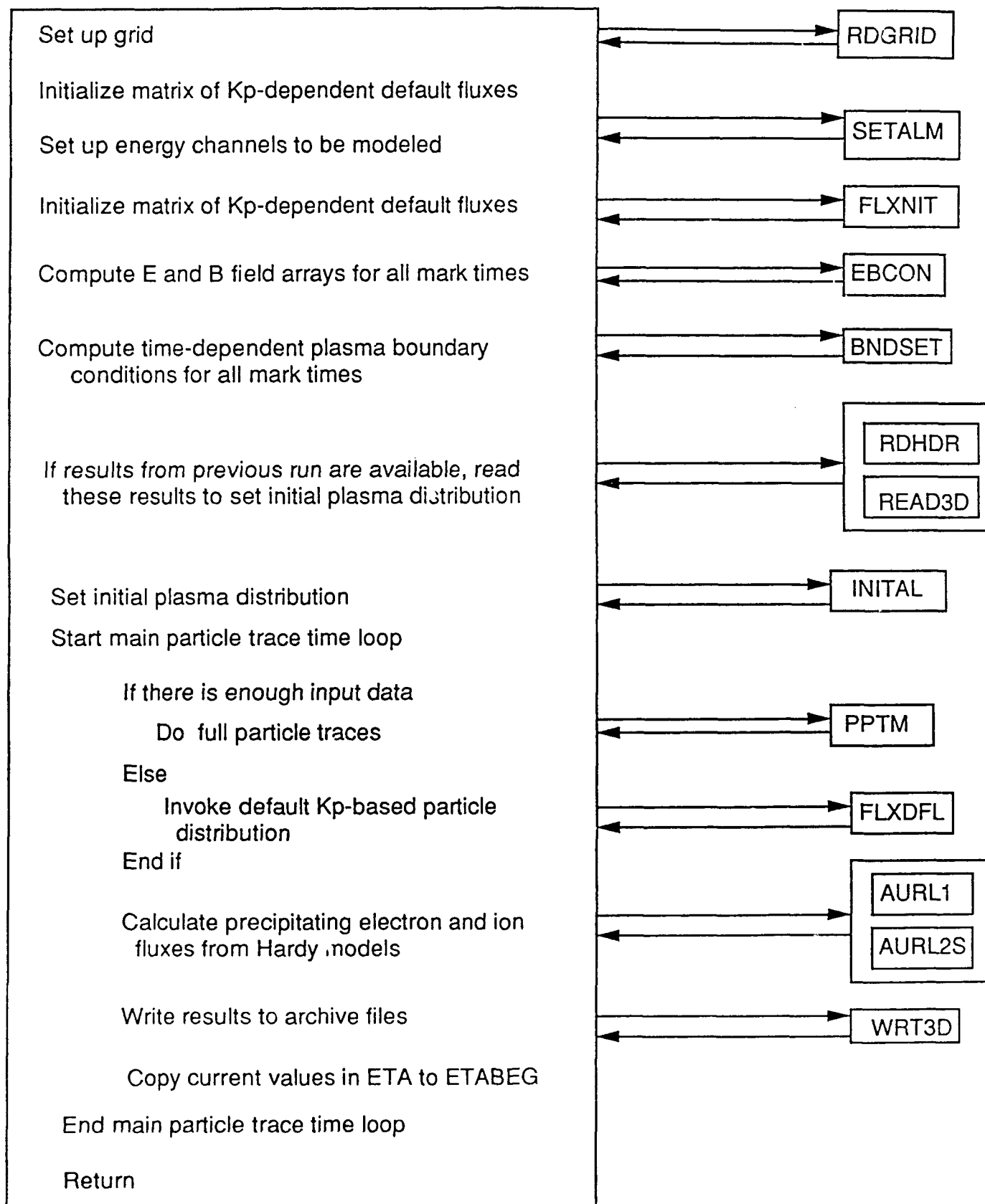
MSMCON

Main MSM control program



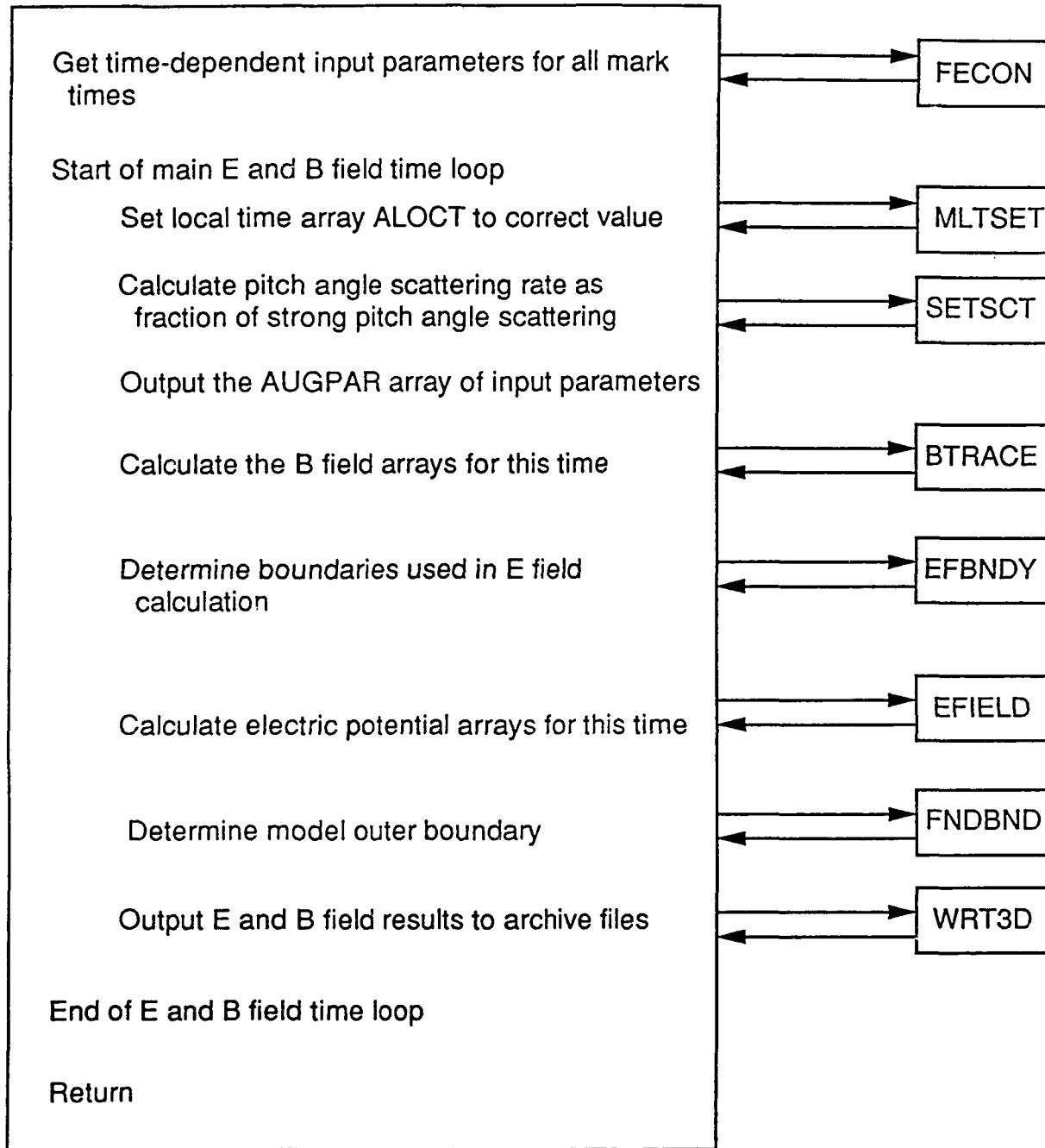
PPTCON

Subroutine that controls the major computations of the MSM.
This routine calls EBCON to construct E and B field matrices at
mark times and then proceeds to trace particle trajectories from
each grid point for each mark time.



EBCON

Calculate E and B field arrays at grid points for mark times



PPTM

Subroutine to trace particle trajectories from each grid point
for each particle species back in time one major time step from
time = TSTART to time = TSTOP. (TSTART > TSTOP)

Calculate reference flux array

Loop over all particle species

Loop over all grid points

Determine normalized time BBT

Set ETA to boundary value of eta for
points outside of boundary

Calculate locations and energies at
TSTART

Start adaptive Runge-Kutta time loop

Move particle a time step and
increment time

Interpolate to find location, flux
tube volume and energy at end
of small RK time step

Calculate loss rate this time step

If time boundary is crossed
(ie., $T < TSTOP$) exit Runge-Kutta
time loop

If spatial boundary is crossed
exit Runge-Kutta time loop

Else go to beginning of Runge-
Kutta time loop

End of Runge-Kutta time loop

Calculate ETA at grid point from
traceback and loss information.
ETA is not allowed to drop below
equilibrium flux value. Loss depends
on threshold for strong pitch angle
scattering.

Calculate precipitating flux and energy

End loop over grid points

End loop over species

Return

SETREF

TNORML

G3NTRP

MOVER

G3NTRP

TCHK

BNDCHK

SYSTEM/SUBSYSTEM SPECIFICATION

SYSTEM/SUBSYSTEM SPECIFICATION - 1

System/Subsystem Specification

2.1 System/Subsystem Description. Attached is a flow chart of the major components of the Magnetospheric Specification Model. The major components of the model are MSMCON, PPTCON, EBCON, and PPTM.

2.2 System/Subsystem Functions.

MSMCON	Main program - sets start, ending, and time steps for the run
PPTCON	Executive routine to control flow of Magnetospheric Specification Model
EBCON	Executive routine to control calculation of electric and magnetic field models
PPTM	Subroutine to trace particle trajectories from each grid point for each particle species back in time from time = TSTART to time = TSTOP

2.2.1 Accuracy and Validity. The Magnetospheric Specification Model produces the most accurate and reliable specifications possible given the available input parameters and the state of magnetospheric science. The standard deviation of the model fluxes and available observed geosynchronous satellite data is output after each run.

2.2.2 Timing. Attached is a report on the throughput time on the IBM 3081 computer. All input data for the run will be acquired from the operational Environmental Database at the beginning of the run as shown on the MSM Control and Data Flow Diagram. All output data will be returned to the Environmental Database. No file interrogation takes place during the run.

4.3 System Data. The following files are necessary to restart the program:

BOxxxxxx	The magnetic field matrices needed for the model run.
COORD	File of values to set up the coordinate system.
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.

SYSTEM/SUBSYSTEM SPECIFICATION - 2

4.3.1 Inputs.

Following is input within the program:

ISTART(1)	Start Year (last two digits)
ISTART(2)	Start Day (julian day)
ISTART(3)	Start Time (seconds)
IEND(1)	End Year (last two digits)
IEND(2)	End Day (julian day)
IEND(2)	End Time (seconds)
IINC(1)	Program time increment (year)
IINC(2)	Program time increment (day)
IINC(3)	Program time increment (seconds)
IRECEB	For restart, record number of previously calculated electric and magnetic field output data to use as input for current run
IRECPT	For restart, record number of previously calculated particle-trace output data to use as input for current run

Following is the input files required, including those to be replace by calls to the environmental data base:

BOxxxxxx	The magnetic field matrices needed for the model run.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).

SYSTEM/SUBSYSTEM SPECIFICATION - 3

SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm ³)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

4.3.2 Outputs. The following files are output:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VNORTH	Northern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VM	Flux tube volume ((Re/nT) ^{-2/3}). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BMIN	Equatorial magnetic field strength (nT). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BNDLOC	Location of outer boundary of detailed particle traces. Output volume - number of longitudinal gridpoints × number of time steps × 4 (bytes)
EFLUX	Precipitating energy flux array (ergs/cm ² -sec). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons)

SYSTEM/SUBSYSTEM SPECIFICATION - 4

	(2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 3 particle species × 4 (bytes)
AUGPAR	Augmented data array for input values.
	Output volume - number of data elements × number of time steps × 4 (bytes)
COLAT	Grid colatitude array (radians).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
ALOCT	Grid local time array (radians eastward from noon).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
FLUX	Flux values for all energy channels at all grid points
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × number of energy channels × 4 (bytes)

4.3.3 Database/Data Bank. The attached chart lists the parameter inputs from the operational Environmental Database and a proposed data transfer method.

4.4 Software Unit Descriptions.

4.4.1 MSMCON - see attached flow chart.

PPTCON	Main control program for model
HIEPAR	High energy electron subroutine, currently a dummy routine pending computer code from Dan Baker, GSFC

4.4.2 PPTCON - see attached flow chart.

RDGRID	Subroutine to read grid system coordinates and calculate essential grid quantities
OUTP	Utility subroutine to print out array information
FLXNIT	Subroutine to calculate Kp-based particle number flux in units of particles/cm ² /s/sr/eV as function of R, energy and Kp
DTIME	Routine which returns current system time
EBCON	Executive routine to control calculation of electric and magnetic field models
EPREAD	Dummy routine to read satellite flux data - to be replaced by call to environmental data base when program is operational
SETALM	Subroutine to read the number of energy channels and their energies at geosynchronous orbit and calculate the energy invariant species needed to run the program

SYSTEM/SUBSYSTEM SPECIFICATION - 5

BNDSET	Subroutine to compute boundary plasma distribution
RDHDR	Subroutine to read the header record of the standard MSM disk file format
INITAL	Subroutine to compute initial plasma distribution
PPTM	Subroutine to trace particle trajectories from each grid point for each particle species back in time from time=TSTART to time=TSTOP
FLXDFL	Subroutine to calculate the energy-dependent ETA array from Kp and the location of the grid points by interpolating the empirical FLXMAT array - this routine is used as the MSM default model when full particle traces are not done
AURL1	Subroutine to determine default Hardy precipitating electron flux
AURL2S	Subroutine to determine default Hardy precipitating ion flux
PWRCAL	Subroutine to estimate precipitating electron energy flux poleward of the main MSM modeling region by comparing flux within modeling region with Hardy statistical values
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format

4.4.3 EBCON - see attached flow chart.

FECON	Subroutine to obtain time-normalized observational values
MLTSET	Subroutine to calculate time dependent ALOCT of rotating grid points
SETSCT	Subroutine to set species dependent pitch angle scattering efficiency
BTRACE	Subroutine to calculate magnetic field models appropriate for geophysical conditions as specified by standoff distance, equatorward edge of the auroral zone, Dst, tail collapse, and tilt angle
EFBN DY	Subroutine to calculate latitudes and widths of electric field model boundaries 2 and 3
EFIELD	Subroutine to return values of the potential on the northern hemisphere grid (VNORTH), on the southern hemisphere grid (VSOUTH), and the average of the two hemispheres (V)
OUTP	Utility subroutine to print out array information
FNDBND	Subroutine to compute time dependent outer boundary location
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
VMLSET	Subroutine to calculate VMLOSS array
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format

4.4.4 PPTM - see attached flow chart.

SETREF	Subroutine to set upper limit reference flux
OUTP	Utility routine to print our array information
TNORML	Utility function subprogram to calculate run-normalized time
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
WKRATE	Function subprogram to evaluate weak precipitation loss rate
MOVER	Subroutine to advance one particle location one time step using 4th-order Runge-Kutta algorithm with 5th order correction

SYSTEM/SUBSYSTEM SPECIFICATION - 6

MODFIX	Subroutine to adjust BBJ to be between 2 and JDIM-1
TCHK	Subroutine to check whether time boundary was crossed
BNDCHK	Subroutine to determine whether particle has crossed boundary of calculation
FLXVAL	Function subprogram to calculate Kp-dependent flux value a $R=RR$ for particles of energy ENRG and $Kp=FKP$ by interpolating FLXMAT array

PARAMETER	NAME	UNITS	IO	Time Tag	Geo. Latitude	Geo. Longitude	Altitude	Mag. Local Time
-----------	------	-------	----	----------	---------------	----------------	----------	-----------------

INPUT ONLY:

Kp	FKP	NONE	I	X				
Dst	DST	NANOTESLA	I	X				
C2	C2	NONE	I	X				
LOW LAT AURORAL BOUNDARY	DLATAZ	DEGREES	I	X			X	X
EQ. EDGE R1 CURRENTS	EDGER1	DEGREES	I	X			X	X
EQ. EDGE R2 CURRENTS	EDGER2	DEGREES	I	X			X	X
POLAR CAP POTENTIAL	PCP	KILOVOLTS	I	X				
POLAR CAP POTENTIAL PATTERN	XIPATT	NONE	I	X				
POLAR CAP BOUNDARY LATITUDE	PCBND	T.B.D.	I	X				
PRECIPITATION POWER INDEX	P	NONE	I	X				
GEO MAG X COMP.	GEOMGX	NANOTESLA	I	X				X
GEO MAG Y COMP.	GEOMGY	NANOTESLA	I	X				X
GEO MAG Z COMP.	GEOMGZ	NANOTESLA	I	X				X
GEO ELECTRON FLUX - CHANNEL(I)	EFLX(I)	ELEC/CM ² -s	I	X				X
GEO ION FLUX - CHANNEL(I)	GFLX(I)	IONS/CM ² -s	I	X				X
SOLAR WIND VELOCITY	SWVEL	KM/s	I	X	X	X	X	X
SOLAR WIND DENSITY	SWDEN	PROT/CM ³	I	X	X	X	X	X
IMF X COMPONENT	XIMFBX	NANOTESLA	I	X	X	X	X	X
IMF Y COMPONENT	XIMFBY	NANOTESLA	I	X	X	X	X	X
IMF Z COMPONENT	XIMFBZ	NANOTESLA	I	X	X	X	X	X

Data Transfer Method

1. MSM will access the environmental database input data files by calling a subroutine, **INDATA**.
2. The call will be of the form:
CALL INDATA(parm-name, STARTT, ENDT, NDIM, DARRY, NUMNUM)
 Where:
parm-name is a valid parameter name from the MSM input parameter list.
STARTT is the starting time for this data request.
ENDT is the ending time for this data request.
NDIM is the horizontal dimension of DARRY and the maximum number of data points that may be returned.
DARRY is a two dimensional array to be retrieved from the database. The form of DARRY is shown on the next page. The size of DARRY will be NDIM by 7.
NUMNUM is the number of data values placed in DARRY by INDATA.
3. The system controller will periodically update the input data files.
4. Rice does not plan to write **INDATA**.
5. All parameters will be in single precision, floating point format.
6. Output data will be passed from MSM to the environmental database via a subroutine, **OTDATA**.
7. The call to **OTDATA** will be defined similarly to **INDATA**.
8. Rice does not plan to write **OTDATA**.

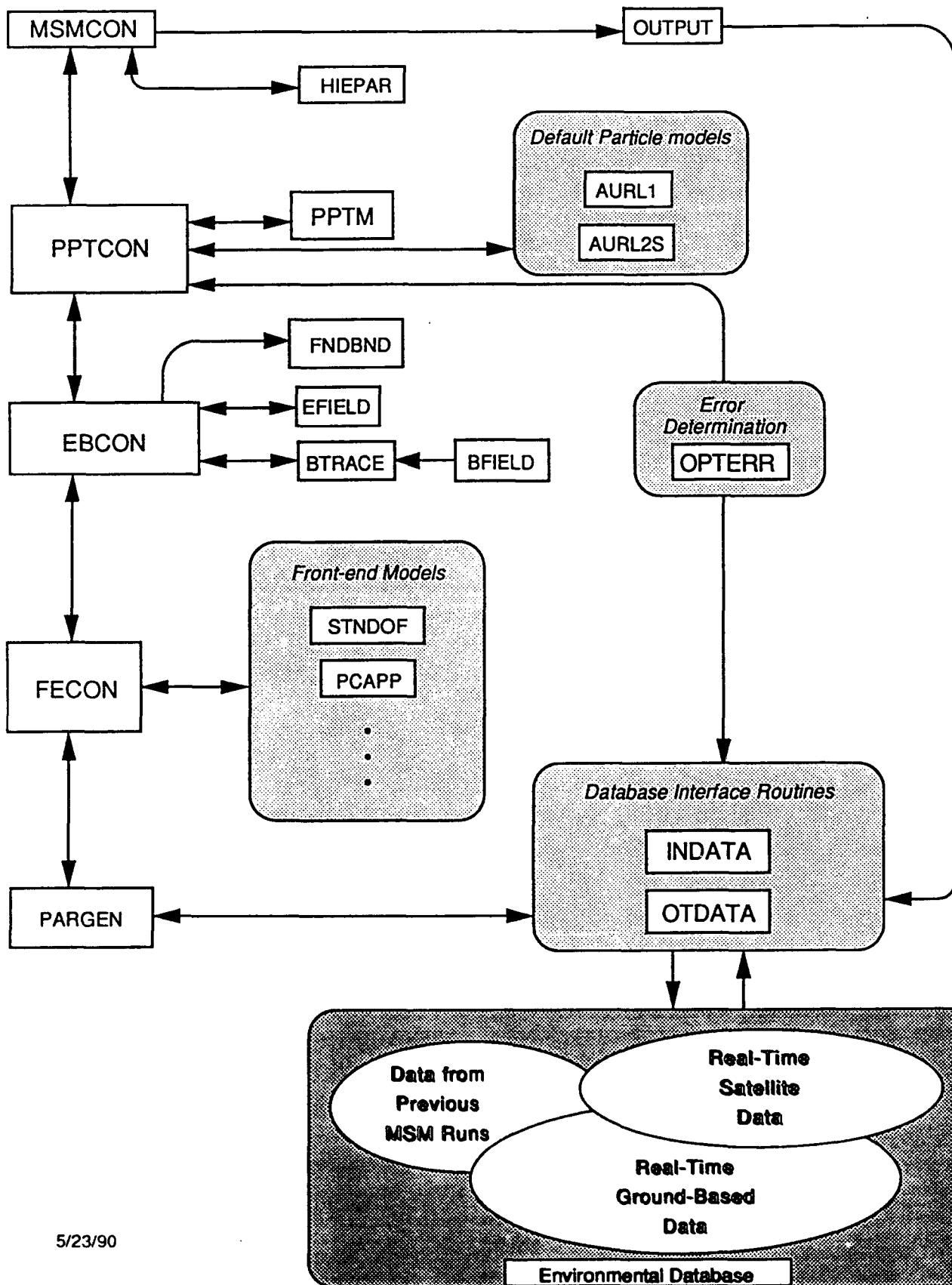
Format for DARRY

NDIM

→

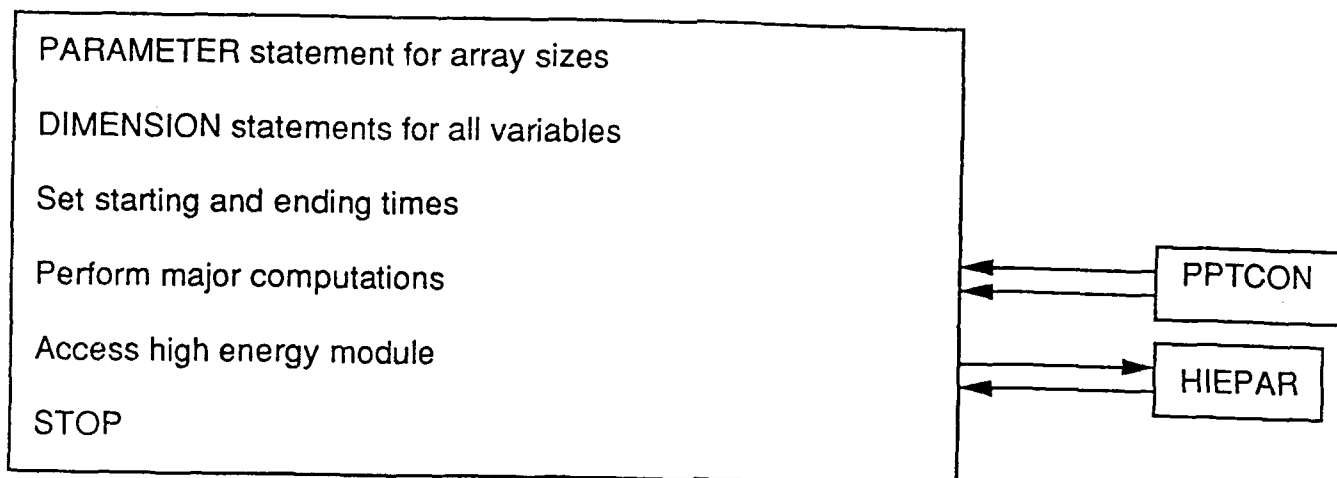
7 ↓	Data value at t1	Data value at t2	Data value at t3	Etc.
	Time tag at t1	Time tag at t2	Time tag at t3	
	Latitude	Latitude	Latitude	
	Longitude	Longitude	Longitude	
	Altitude	Altitude	Altitude	
	mag. local time	mag. local time	mag. local time	

MSM Control & Data Flow Diagram



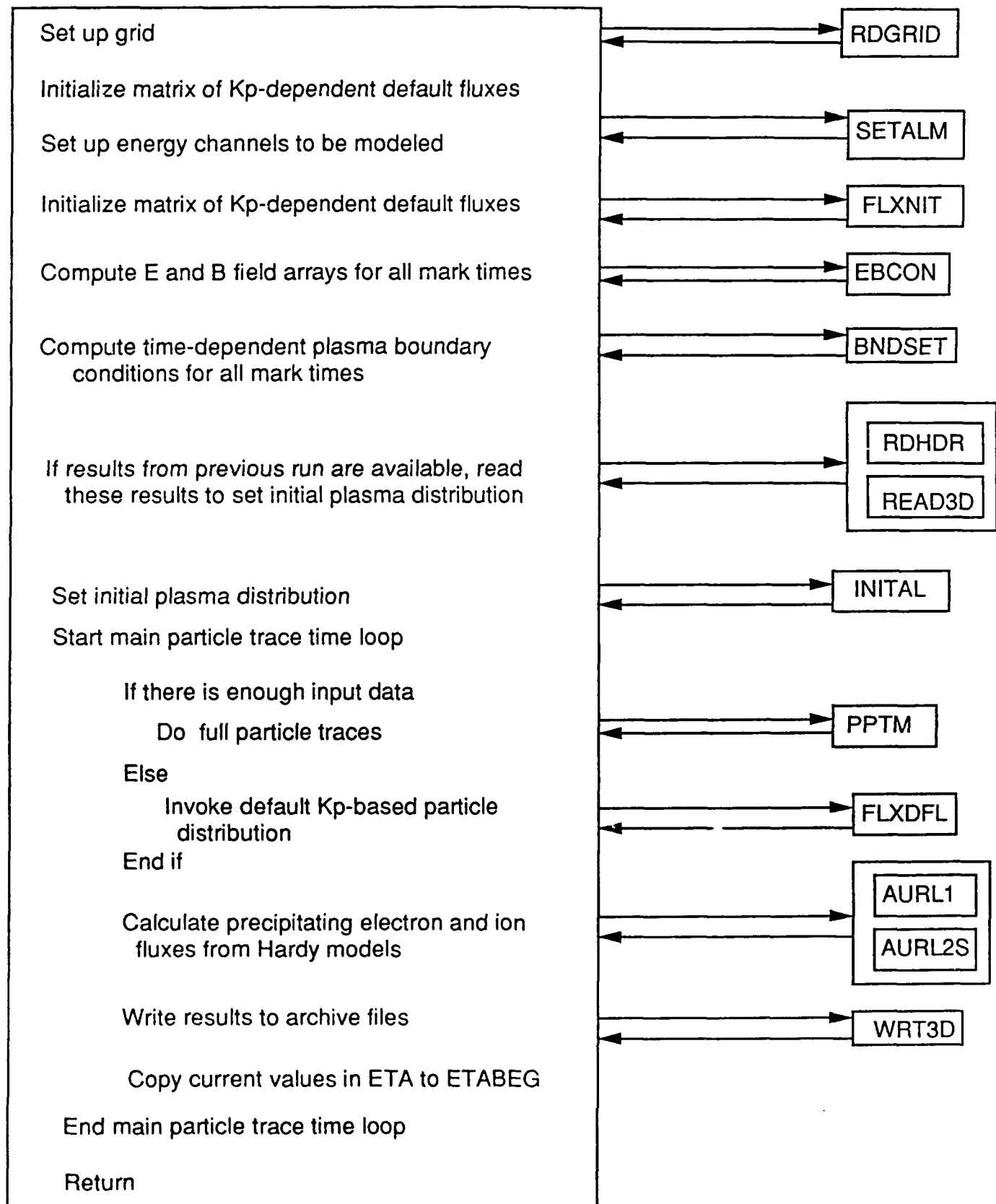
MSMCON

Main MSM control program



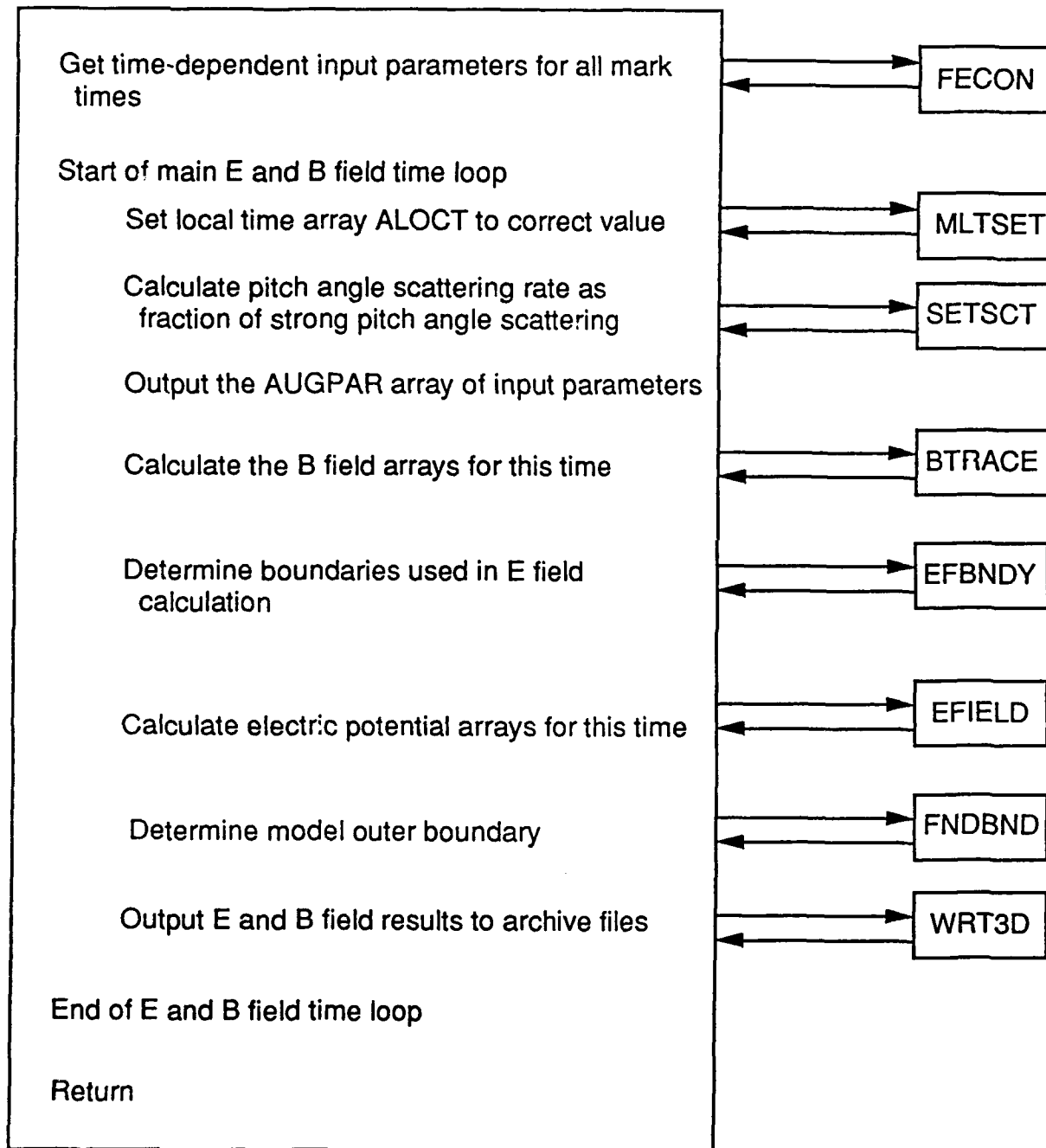
PPTCON

Subroutine that controls the major computations of the MSM.
This routine calls EBCON to construct E and B field matrices at mark times and then proceeds to trace particle trajectories from each grid point for each mark time.



EBCON

Calculate E and B field arrays at grid points for mark times



PPTM

Subroutine to trace particle trajectories from each grid point
for each particle species back in time one major time step from
time = TSTART to time = TSTOP. (TSTART > TSTOP)

Calculate reference flux array

Loop over all particle species

Loop over all grid points

Determine normalized time BBT

Set ETA to boundary value of eta for
points outside of boundary

Calculate locations and energies at
TSTART

Start adaptive Runge-Kutta time loop

Move particle a time step and
increment time

Interpolate to find location, flux
tube volume and energy at end
of small RK time step

Calculate loss rate this time step

If time boundary is crossed
(ie., $T < TSTOP$) exit Runge-Kutta
time loop

If spatial boundary is crossed
exit Runge-Kutta time loop

Else go to beginning of Runge-
Kutta time loop

End of Runge-Kutta time loop

Calculate ETA at grid point from
traceback and loss information.
ETA is not allowed to drop below
equilibrium flux value. Loss depends
on threshold for strong pitch angle
scattering.

Calculate precipitating flux and energy

End loop over grid points

End loop over species

Return

SETREF

TNORML

G3NTRP

MOVER

G3NTRP

TCHK

BNDCHK

SOFTWARE UNIT SPECIFICATION

SOFTWARE UNIT SPECIFICATION - 1

Software Unit Specification

2.2 Software Unit Functions. See the attached pages for the subroutine names, functions and calling environment.

2.2.1 Accuracy and Validity. The Magnetospheric Specification Model produces the most accurate and reliable specifications possible given the available input parameters and the state of magnetospheric science. The standard deviation of the model fluxes and available observed geosynchronous satellite data is output after each run.

2.2.2 Timing. Attached is a report on the throughput time on the IBM 3081 computer. All input data for the run will be acquired from the operational Environmental Database at the beginning of the run as shown on the MSM Control and Data Flow Diagram. All output data will be returned to the Environmental Database. No file interrogation takes place during the run.

4.3 Inputs.

Following is input within the program:

ISTART(1)	Start Year (last two digits)
ISTART(2)	Start Day (julian day)
ISTART(3)	Start Time (seconds)
IEND(1)	End Year (last two digits)
IEND(2)	End Day (julian day)
IEND(2)	End Time (seconds)
IINC(1)	Program time increment (year)
IINC(2)	Program time increment (day)
IINC(3)	Program time increment (seconds)
IRECEB	For restart, record number of previously calculated electric and magnetic field output data to use as input for current run
IRECPT	For restart, record number of previously calculated particle-trace output data to use as input for current run

Following is the input files required, including those to be replaced by calls to the environmental data base:

BOxxxxxx	The magnetic field matrices needed for the model run.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.

SOFTWARE UNIT SPECIFICATION - 2

COORD	File of values to set up the coordinate system.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm ³)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

4.3 Outputs. The following files are output:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VNORTH	Northern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VM	Flux tube volume ((Re/nT) ^{-2/3}). Output volume - number of latitudinal gridpoints × number of longitudinal

SOFTWARE UNIT SPECIFICATION - 3

BMIN	<p>gridpoints \times number of time steps \times 4 (bytes)</p> <p>Equatorial magnetic field strength (nT).</p> <p>Output volume - number of latitudinal gridpoints \times number of longitudinal</p>
XMIN	<p>gridpoints \times number of time steps \times 4 (bytes)</p> <p>GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).</p> <p>Output volume - number of latitudinal gridpoints \times number of longitudinal</p>
YMIN	<p>gridpoints \times number of time steps \times 4 (bytes)</p> <p>GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).</p> <p>Output volume - number of latitudinal gridpoints \times number of longitudinal</p>
ZMIN	<p>gridpoints \times number of time steps \times 4 (bytes)</p> <p>GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).</p> <p>Output volume - number of latitudinal gridpoints \times number of longitudinal</p>
BNDLOC	<p>gridpoints \times number of time steps \times 4 (bytes)</p> <p>Location of outer boundary of detailed particle traces.</p> <p>Output volume - number of longitudinal gridpoints \times number of time steps \times 4 (bytes)</p>
EFLUX	<p>Precipitating energy flux array (ergs/cm²-sec).</p> <p>Output volume - number of latitudinal gridpoints \times number of longitudinal</p>
FLXSUM	<p>gridpoints \times number of time steps \times number of energy channels \times 4 (bytes)</p> <p>Total precipitation energy flux array (integrated over species for (1) electrons (2) H⁺ and (3) O⁺) (ergs/cm²-sec).</p> <p>Output volume - number of latitudinal gridpoints \times number of longitudinal</p>
EAVG	<p>gridpoints \times number of time steps \times 3 particle species \times 4 (bytes)</p> <p>Average precipitating electron energy for (1) electrons, (2) H⁺ and (3) O⁺ (eV).</p> <p>Output volume - number of latitudinal gridpoints \times number of longitudinal</p>
AUGPAR	<p>gridpoints \times number of time steps \times 3 particle species \times 4 (bytes)</p> <p>Augmented data array for input values.</p> <p>Output volume - number of data elements \times number of time steps \times 4 (bytes)</p>
COLAT	<p>Grid colatitude array (radians).</p> <p>Output volume - number of latitudinal gridpoints \times number of longitudinal</p>
ALOCT	<p>gridpoints \times number of time steps \times 4 (bytes)</p> <p>Grid local time array (radians eastward from noon).</p> <p>Output volume - number of latitudinal gridpoints \times number of longitudinal</p>
FLUX	<p>gridpoints \times number of time steps \times 4 (bytes)</p> <p>Flux values for all energy channels at all grid points</p>

SOFTWARE UNIT SPECIFICATION - 4

Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times number of energy channels \times 4 (bytes)

MSM Routines by Function

Control Routines

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
MSMCON	Main program Sets start and ending times and time step for run.	not applicable	PPTCON HIEPAR
PPTCON	Executive routine to control flow of MSM. E and B field models are computed in EBCON; particle traces are done in PPTM.	MSMCON	RDGRID OUTP FLXNIT DTIME EBCON EPREAD SETALM BNDSET RDHDR INITAL PPTM FLXDFL AURL1 PWRCAL WRT3D
EBCON	Executive routine to control calculation of electric and magnetic field models.	PPTCON	FECON MLTSET SETSCT BTRACE EFBNDY EFIELD OUTP FNDBND G3NTRP VMLSET WRT3D
HIEPAR	Subroutine to give energetic particle results.	MSMCON	presently dummy

Magnetic Field Routines

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
BTRACE	Subroutine to calculate B field models appropriate for geophysical conditions as specified by FSTOFF, FEQEDG, FDST, FCLPSE, and FTILT.	EBCON	GETMAT RMVBESH BFGYRO
BFGYRO	Subroutine to rotate fixed grid B field arrays to match rotating coordinate system.	BTRACE	none
FNDBRK	Subroutine to find the proper B field array indices that bracket the magnetic field input parameters.	GETMAT	none
GETMAT	Subroutine to find the B field matrices needed for interpolation in order to represent current conditions.	BTRACE	MXINIT FNDBRK LOADBM ZEROBM OUTP
LOADBM	Subroutine to load individual B-matrices from the B-supermatrix into the work B-matrices.	GETMAT	none
MXINIT	Subroutine to initialize the array MEXIST to reflect the B-matrices that exist in the B-super matrix.	GETMAT	none
RMVBESH	Subroutine to extrapolate B-field matrices "into the bushes", ie., into extremely stretched or open regions.	BTRACE	none
ZEROBM	Subroutine to zero out an individual B matrix within the working B matrices. This is done when an individual matrix needed in the interpolation does not exist in the computed B-supermatrix.	GETMAT	none

Electric Field Routines

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
EFIELD	Subroutine to return values of the potential on the northern hemisphere grid (VNORTH), on the southern hemisphere grid (VSOUTH), and the average of the two hemispheres (V).	EBCON	EMODEL
AURORA	Returns potential at a specific location in auroral region.	EMODEL	none
EFBNDY	Subroutine to calculate latitudes and widths of efield bndys 2 and 3.	EBCON	EFLOC G3NTRP
EFLOC	Subroutine to find grid location (BI,BJ) of physical location (RVAL,XLT).	EFBNDY	PFIX
EMODEL	Returns the electric potential arrays VNORTH, VSOUTH, and V at all grid pts.	EFIELD	INPUT THET EPOT LOW AURORA REG1
EPOT	Using scaled Heppner-Maynard model returns potential at a specific pt.	EMODEL	none
FNDBND	Subroutine to compute time dependent outer boundary location.	EBCON	THET
INPUT	Get input needed for computation of Heppner-Maynard model.	EMODEL	none
LOW	Returns potential at a specific location in low latitude region.	EMODEL	none
REG1	Returns potential at a specific location in region 1.	EMODEL	none
THET	Function that gives eqn for an E field boundary ellipse in flat polar coordinates.	EMODEL FNDBND PWRCAL	none

Particle Trace Routines

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
PPTM	Subroutine to trace particle trajectories from each grid pt for each particle species back in time from time=TSTART to time=TSTOP.	PPTCON	SETREF OUTP TNORML G3NTRP WKRATE MOVER MODFIX TCHK BNDCHK FLXVAL
BNDCHK	Subroutine to determine whether particle has crossed bndy of calculation.	PPTM	TNORML
BNDSET	Subroutine to compute boundary plasma distribution.	PPTCON	none
CFACT	Function subprogram to compute correction factor based on equilibrium eta values.	not called	none
DVEFDI	Function subprogram to compute the I derivative of the effective potential at grid pt (I,J) at normalized time BT.	VLOCTY	none
DVEFDJ	Function subprogram to compute the J derivative of the effective potential at grid pt (I,J) at normalized time BT.	VLOCTY	none
FLXCAL	Subroutine to compute Kp-dependent flux at L=3,4,6,6, and 13 for electrons of given energy for a given Kp condition.	FLXNIT	none
FLXNIT	Subroutine to calculat Kp based particle number flux in units of particles/cm**2/s/sr/eV as function of r, energy, and Kp.	PPTCON	FLXCAL THRCAL
FLXVAL	Function subprogram to calculate Kp-dependent flux value at R=RR for particles of energy ENRG and KP=FKP by interpolating FLXMAT array.	PPTM BNDSET INITAL SETREF FLXDFL	G3NTRP

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
FLXTRP	Function subprogram to return LOG10 of empirical flux value interpolated between values at L=3,4,6.6, and 13 Re.	FLXNIT	none
INITAL	Subroutine to compute initial plasma distribution	PPTCON	FLXVAL
MOVER	Subroutine to advance one particle location one time step using 4th-order Runge-Kutta algorithm with 5th order correction.	PPTM	RK4
PWRCAL	Subroutine to estimate precipitating electron energy flux poleward of the main MSM modeling region by comparing flux within modeling region with Hardy statistical values.	PPTCON	THET
RK4	4th order Runge-Kutta routine	MOVER	VLOCTY
RLYONS	Function subprogram to evaluate Lyons-based weak loss rate.	WKRATE	none
SETALM	Subroutine to read the number of energy channels and their energies at geosynchronous orbit and calculate the energy invariant species needed to run the program.	PPTCON	none
SETREF	Subroutine to set upper limit reference flux.	PPTM	FLXVAL TNORML
SETSCT	Subroutine to set species dependent pitch angle scattering efficiency.	EBCON	none
TCHK	Subroutine to check whether time boundary was crossed.	PPTM	G3NTRP
THRCAL	Subroutine to calculate threshold flux at R=3, 4, 6.6, and 13 RE for KP=FKP and LOG(energy)=FNRGLG.	FLXNIT	none
VLOCTY	Subroutine to compute I and J components of particle velocity with energy invariant AALAM at location (BBI,BBJ) at normalized time BT.	RK4	G3NTRP

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
VMLSET	Subroutine to calculate VMLOSS array.	EBCON	none
WKRATE	Function subprogram to evaluate weak precipitation loss rate.	PPTM	RLYONS

Input and Output Data Handling

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
FECON	Subroutine to obtain time-normalized observational values.	EBCON	PARGEN
PARGEN	Subroutine to obtain values from the environmental data base and interpolate or extrapolate as appropriate to provide data at a normalized time.	FECON	INDATA DFLTVL SMOOTH DTXIPT DTNTRP TIMINC
DTNTRP	Subroutine to return interpolation values for the N data pairs of (XA(I),YA(I)), where I=1,NA.	PARGEN	none
DTXIPT	Subroutine to return interpolation values for XIPATT.	PARGEN	none
INDATA	Subroutine to read in data for the MSM during development and testing. This routine will be replaced in the operational MSM with an interface routine to the AWS environmental database.	PARGEN	none
OUTPUT	Subroutine to output MSM results to data base.	MSMCON	presently a dummy routine.
SMOOTH	Subroutine to check and extract data from DARRAY into XA and YA.	PARGEN	TCONV2
STNDOF	Subroutine to calculate the stand-off distance according to a formula from Alpbach (1979).	FECON	none

Default and Front-End Models

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
ADDHDY	Subroutine to calculate the Hardy model electron flux value for a given point.	AURL1	FSUM
AURL1	Subroutine to determine default Hardy precipitating electron flux.	PPTCON	FNDHDY ADDHDY
AURL2S	Subroutine to calculate the default Hardy precipitating ion model.	PPTCON	REGEN EFUN
CLPDFL	Subroutine to determine whether or not to use the collapsed tail version of the magnetic field model.	FECON	none
DSTDFL	Subroutine to calculate default (Kp driven) Dst value.	FECON	none
EFUN	Function subprogram used in evaluation of precipitating ion flux.	AURLS2	none
EQTDFL	Subroutine to calculate default (Kp driven) equatorward edge of auroral zone value.	FECON	none
FLXDFL	Subroutine to calculate the energy-dependent ETA array from Kp and the location of the grid pts by interpolating the empirical FLXMAT array. This routine is used as the MSM default model when full particle traces are not done.	PPTCON	FLXVAL
FNDHDY	Subroutine to find and read the pertinent Fourier coefficients for the Hardy electron model.	AURL1	none
FSUM	Function subprogram that determines the sum of a Fourier series for Hardy model.	ADDHDY	none
PATDFL	Subroutine to return polar-cap convection pattern type on the basis of measurements of IMF By and Bz components if direct measurements of pattern type from DMSP are not available.	FECON	none

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
PCPDFL	Subroutine to calculate default (Kp driven) cross-polar-cap potential value.	FECON	none
REGEN	Subroutine used in evaluation of Hardy model precipitating ion flux.	AURLS2	none
STNDFL	Subroutine to calculate the standoff distance. If solar wind velocity and density data are available, then Subroutine STNDOF does the actual calculation; otherwise, the default is Kp-based.	FECON	STNDOF
TILT	Subroutine to calculate the Earth's tilt angle. Tilt presently defaults to 0.	FECON	none

Utility Routines

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
BLOCK DATA	Sets index of data elements in the PARRAY and AUGPAR arrays. Sets I,J, and energy indices for locations and species to be traced.	Not applicable	Not applicable
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at pt (BI,BJ,BK).	EBCON PPTM TCHK VLOCTY EFBNDY FLXVAL PWRCAL	none
G3TRPA	Utility function subprogram to perform a general 3-d linear interpolation of an angular array.	PPTM	none
MLTSET	Subroutine to calculate time dependent ALOCT of rotating grid pts.	EBCON	none
MODFLX	Subroutine to adjust BBJ to be between 2 and JDIM-1.	PPTM VLOCTY	none
OUTP	Utility subroutine to print out array information.	Utility routine	none
PFIX	Subroutine to adjust modulus of periodic J coordinate.	EFLOC	none
RDGRID	Subroutine to read grid system coordinates and calculate essential grid quantities.	PPTCON	none
RDHDR	Subroutine to read the header record of the standard MSM disk file format.	PPTCON	none
READ3D	Subroutine to read a record from the standard MSM disk file format.	PPTCON	none
TCONV2	Function to take a start year, start decimal day, and current decimal day to produce a value that is the number of seconds from midnight of the start year and day.	SMOOTH	none

<i>Routine</i>	<i>Purpose</i>	<i>Called by</i>	<i>Calls</i>
TCONV3	Function to take a start year, start day, current year, current day, and current seconds of day to produce a value that is the number of seconds from midnight of the start year and day.	PARGEN	none
TIMINC	Subroutine to increment time for next E and B field record.	EBCON PARGEN FECON	none
TNORML	Utility function subprogram to calculate run-normalized time.	PPTM BNDCHK RK4 FNDLOC SETREF	none
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format.	PPTCON EBCON	none

USERS MANUAL

Users Manual

2.5 System Performance. Some factors on which the system performance depends include:

1. Higher energy particles require more time to process. The particles move so rapidly that a smaller time step is required to move their position and the number of time steps required to move the particles on the grid is correspondingly higher.
2. The output files are updated every program time step. If the program is set to run for many time steps, the storage of these files could be a problem. The maximum number of time steps is set within the program to 50.
3. The maximum number of energy channels which can be run is set within the program to 30. The greater the number of channels selected, the longer the run time will be.

2.7 Database/Data Bank. During development the method of input and output is through disk files. These files will no longer be necessary when the Environmental Data Base interface is in place. Attached is a list of the input parameters and units. Also attached is the proposed data transfer method.

2.8 General Description of Inputs, Processing, Outputs.

a. Inputs -

The following parameters are set within the program:

IINC(1)	Program time step increment (year)
IINC(2)	Program time step increment (day)
IINC(3)	Program time increment (seconds)

Following are the input files required, including those to be replaced eventually by calls to the environmental data base:

* denotes the principal operator interfaces to the program

BOxxxxxx The magnetic field matrices needed for the model run.

CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system.
DKTABLE	Loss lifetimes for computing ion loss by charge exchange.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete.
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN*	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete.
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.
MSMIN*	Specify start and stop times, a character list identifier, and sunspot number
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete.
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete.
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete.
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

b. Processing - After the data is gathered from the input files, it goes through Subroutine PARGEN which interpolates or extraplates data as appropriate to provide data a a normalized run-time. Subroutine FECON then calculates the physical parameters needed by the model for each run time label. When the Environmental Database interface is operational, output from the model will then return to the Environmental Data Base through Subroutine OUTPUT. (At present OUTPUT) is only a place-holding stub.) The MSM Control and Data Flow Diagram is attached.

c. Outputs. The following files are output:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output for each time label.
VNORTH	Northern hemisphere electric potential distribution (Volts). Output for each time label.
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output for each time label
VM	Flux tube volume $((Re/nT)^{-2/3})$. Output for each time label
BMIN	Equatorial magnetic field strength (nT). Output for each time label
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label
BNDLOC	Location of outer boundary of detailed particle traces. Output for each time label
EFLUX	Precipitation energy flux array (ergs/cm ² -sec). Output for each time label
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec). Output for each time label
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV). Output for each time label
AUGPAR	Augmented data array for input values. Output for each time label
COLAT	Grid colatitude array (radians). Output for each time label
ALOCT	Grid local time array (radians eastward from noon). Output for each time label
FLUX	Flux values for all energy channels at all grid points. Output for each time label for each energy channel

3.2 Input Requirements.

Start and stop times are specified to the program through the 5 line MSMIN file as year (eg., 1988), day (eg., 112 for the 112 th day of the year), and time in seconds from the beginning of the start day (eg., 43200 for noon). For example, the following configuration of the MSMIN file will run the

MSM from 1800 on the 112th day of 1988 through 1830 on the same day. This is a fresh startup of the program (direct access record number is zero) and the sunspot number is 0.0.

1988 112 64800	'start year, start day, start time in seconds'
1988 112 66600	'end year, end day, end time in seconds'
0	'0 signifying new start'
'30 minute test run'	'up to 80 character run identification'
0.0	'sunspot number'

The time step increment is set in the program through the IINC vector.

IINC(1) = Increment Year
 IINC(2) = Increment Day
 IINC(3) = Increment Time (seconds)

If the run is to be a restarted using previous data, the direct access record number for restarting is also read from the MSMIN file. For example, the program is currently set up to run year 1988, day 112, hour 18 (64800 seconds) through year 1988, day 112, hour 1830 (66000 seconds) in 15 minute (900 second) increments. Since it is a new run the start record number is set to zero in MSMIN.

If after running this interval you want to restart the model at hour 1830 and continue through hour 20, the MSMIN input file would be:

```
1988 112 66600
1988 112 72000
3
'MSM run restart from 1830 on day 112'
0.0
```

The model would go back, pick up the original 5 records and continue the run from hour 19 to hour 20.

Input file ENCHAN controls how many and which energy channels are calculated. The file set up is:

Number of energy channels to be modeled (maximum 30, integer)
 Particle type 1 (integer), Energy 1 (floating point)
 Particle type 2 (integer), Energy 2 (floating point)

 Particle type n (integer), Energy n (floating point)

The acceptable particle type input is:

1	electrons
2	H ⁺
3	O ⁺

The energy is at geosynchronous orbit in keV. The energies must be in ascending order, with particle types grouped together, which are checked within the program.

For example,

4	
1	40
1	55
2	10
2	15

This is to model 4 energy channels, the first two are electrons of 40 and 55 keV at geosynchronous orbit. The second two are H⁺ ions of 10 and 15 keV energy at geosynchronous orbit.

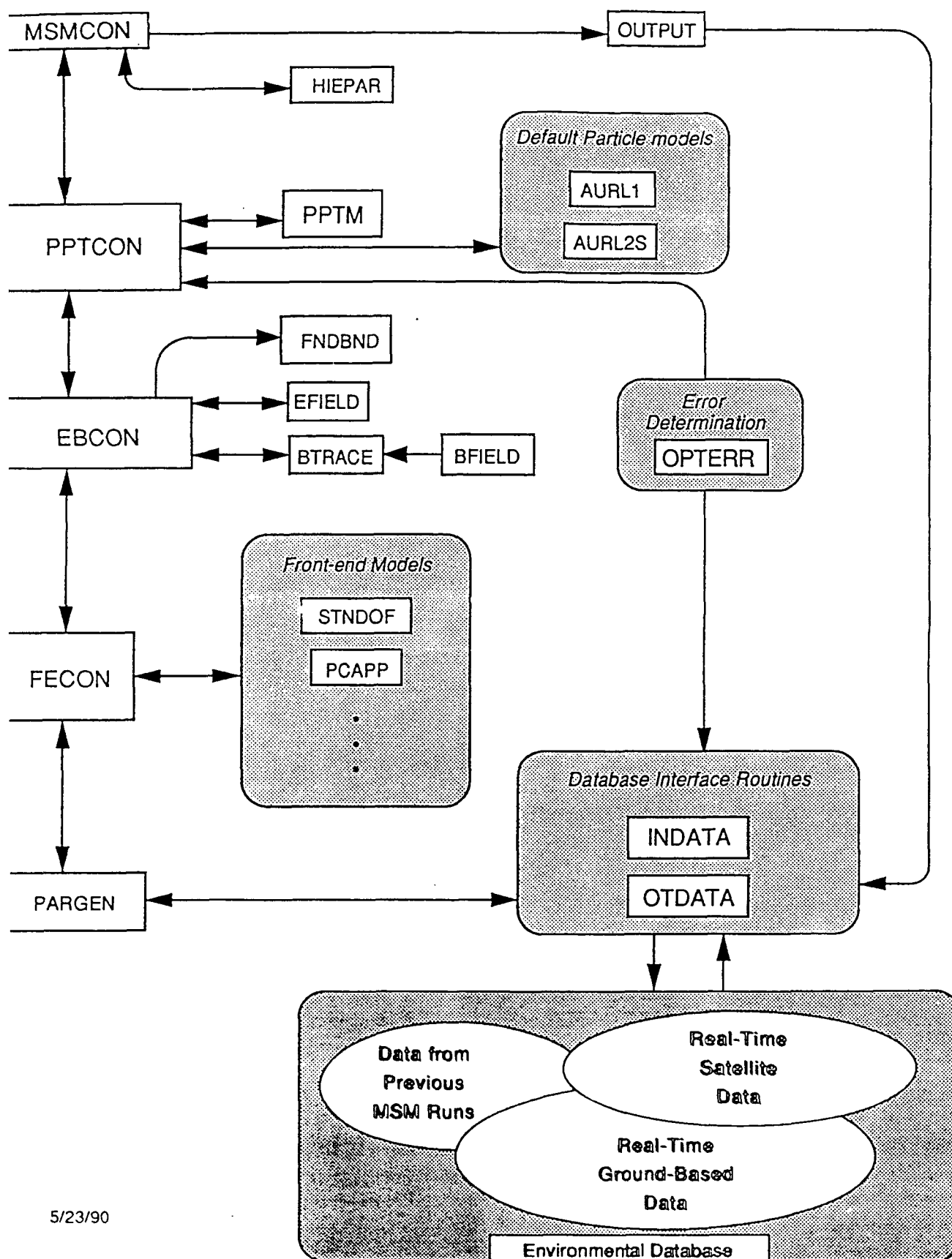
3.3 Output Requirements. The following files are output. When the Environmental Data Base is complete, these files will be stored there. The file of immediate use to forecasters is the FLUX data file which contains the particle flux information for the desired energy channels at all points on the model grid. The other data files will be needed for post-analysis of the event.

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output for each time label.
VNORTH	Northern hemisphere electric potential distribution (Volts). Output for each time label.
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output for each time label.
VM	Flux tube volume $((R_e/nT)^{-2/3})$. Output for each time label.
BMIN	Equatorial magnetic field strength (nT). Output for each time label.
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label.
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label.
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output for each time label.
BNDLOC	Location of outer boundary of detailed particle traces. Output for each time label.
EFLUX	Precipitation energy flux array (ergs/cm ² -sec). Output for each time label.
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec). Output for each time label.
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV). Output for each time label.

AUGPAR
COLAT
ALOC
FLUX

Augmented data array for input values. Output for each time label
Grid colatitude array (radians). Output for each time label
Grid local time array (radians eastward from noon). Output for each time label
Flux values for all energy channels at all grid points. Output for each time label
for each energy channel

MSM Control & Data Flow Diagram



PARAMETER	NAME	UNITS	IO	Time Tag	Geo. Latitude	Geo. Longitude	Altitude	Mag. Local Time
-----------	------	-------	----	----------	---------------	----------------	----------	-----------------

INPUT ONLY:

Kp	FKP	NONE	I	X				
Dst	DST	NANOTESLA	I	X				
C2	C2	NONE	I	X				
LOW LAT AURORAL BOUNDARY	DLATAZ	DEGREES	I	X			X	X
EQ. EDGE R1 CURRENTS	EDGER1	DEGREES	I	X			X	X
EQ. EDGE R2 CURRENTS	EDGER2	DEGREES	I	X			X	X
POLAR CAP POTENTIAL	PCP	KILOVOLTS	I	X				
POLAR CAP POTENTIAL PATTERN	XIPATT	NONE	I	X				
POLAR CAP BOUNDARY LATITUDE	PCBND	T.B.D.	I	X				
PRECIPITATION POWER INDEX	P	NONE	I	X				
GEO MAG X COMP.	GEOMGX	NANOTESLA	I	X				X
GEO MAG Y COMP.	GEOMGY	NANOTESLA	I	X				X
GEO MAG Z COMP.	GEOMGZ	NANOTESLA	I	X				X
GEO ELECTRON FLUX - CHANNEL(I)	EFLX(I)	ELEC/cm ² .s	I	X				X
GEO ION FLUX - CHANNEL(I)	GFLX(I)	IONS/cm ² .s	I	X				X
SOLAR WIND VELOCITY	SWVEL	KM/s	I	X	X	X	X	X
SOLAR WIND DENSITY	SWDEN	PROT/cm ³	I	X	X	X	X	X
IMF X COMPONENT	XIMFBX	NANOTESLA	I	X	X	X	X	X
IMF Y COMPONENT	XIMFYZ	NANOTESLA	I	X	X	X	X	X
IMF Z COMPONENT	XIMFBZ	NANOTESLA	I	X	X	X	X	X

END USERS MANUAL

End Users Manual

2.1.2 Performance. Some factors on which the system performance depends include:

1. Higher energy particles require more time to process. The particles move so rapidly that a smaller time step is required to move their position and the number of time steps required to move the particles on the grid is correspondingly higher.
2. The output files are updated every program time step. If the program is set to run for many time steps, the storage of these files could be a problem. The maximum number of time steps is set within the program to 50.
3. The maximum number of energy channels which can be run is set within the program to 30. The greater the number of channels selected, the longer the run time will be.

COMPUTER OPERATION MANUAL

COMPUTER OPERATION MANUAL - 1

Computer Operation Manual

2.1 System Application. The Magnetospheric Specification Model was developed as a set of algorithms to be developed into a real-time program for use in the Space Forecast Center operated by the Air Weather Service of the U.S. Air Force. It is intended to help the Center personnel perform their mission of providing information, particularly during magnetospheric disturbances, to "customers" who operate spacecraft. The output of the model includes (but is not limited to):

- 1) Fluxes of electrons and ions in the inner plasma sheet and the geosynchronous orbit region.
- 2) Energy fluxes and characteristic energies of electrons precipitated into the auroral ionosphere.

2.4 Information Inventory.

2.4.1 Resource Inventory.

Following are the files accessed by the MSM, including those to be eventually replaced by calls to the environmental data base. Those required for a restart are so labelled.

Input data sets

* denotes principal mode of operator control of program

BOxxxxxx	The magnetic field matrices needed for the model run. Required for restart.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system. Required for restart.
DKTABLE	File of decay times for computing ion loss by charge exchange.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model. Required for restart.
ENCHAN*	Input values for the number of energy channels, type and energy of particles to be traced. Required for restart.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete. Required for restart.
HARDY	Coefficients used by the Hardy precipitating electron model. Required for restart.
IONENG	Coefficients used as input for the ion precipitating model. Required for restart.

COMPUTER OPERATION MANUAL - 2

IONNUM	Coefficients used as input for the ion precipitating model. Required for restart.
MSMIN*	Primary input file to control run. Provides start and end times for run, a character string identifier for run, and sun spot number. Required for restart.
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm ³)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

Output data sets

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VNORTH	Northern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VSOUTH	Southern hemisphere electric potential distribution (Volts). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VM	Flux tube volume $((Re/nT)^{-2/3})$. Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
BMIN	Equatorial magnetic field strength (nT). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re). Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).

COMPUTER OPERATION MANUAL - 3

	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
BNDLOC	Location of outer boundary of detailed particle traces.
	Output volume - number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
EFLUX	Precipitating energy flux array (ergs/cm ² -sec).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times number of energy channels \times 4 (bytes)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 3 particle species \times 4 (bytes)
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 3 particle species \times 4 (bytes)
AUGPAR	Augmented data array for input values.
	Output volume - number of data elements \times number of time steps \times 4 (bytes)
COLAT	Grid colatitude array (radians).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
ALOCT	Grid local time array (radians eastward from noon).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
FLUX	Flux values for all energy channels at all grid points. Needed for restart.
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times number of energy channels \times 4 (bytes)

2.4.2 Report Inventory. A print file of arrays of different parameters is currently generated.

TEST PLAN

Test Plan

2. DEVELOPMENT ACTIVITY

2.1 Statement of Pretest Activity. During the development phase the model output was compared to observed data for 2 events - CDAW6 March 22, 1978 interval and April ~~22-24~~, 1988 interval.
21-23

2.2 Pretest Activity Results. Results of the CDAW6 interval are discussed in "Chapter IV - Testing the Model Using the CDAW6 Event" of B. Hausman's Masters Thesis (included as an appendix to this test plan.) MSM results are compared quantitatively with observations for the April ~~22-24~~, 1988 period in the Rice MSM final report.
21-23

3.1 System Description.

Start and stop times and start direct-access record number are provided in input file MSMIN.

Following are the input files required, including those to be replaced by calls to the environmental data base:

BOxxxxxx	The magnetic field matrices needed for the model run.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system.
DKTABLE	Decay times for computing charge-exchange loss of ions
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model.

PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm ³)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.

Following are the output files:

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts).
VNORTH	Northern hemisphere electric potential distribution (Volts).
VSOUTH	Southern hemisphere electric potential distribution (Volts).
VM	Flux tube volume $((Re/nT)^{-2/3})$.
BMIN	Equatorial magnetic field strength (nT).
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re).
BNDLOC	Location of outer boundary of detailed particle traces.
EFLUX	Precipitating energy flux array (ergs/cm ² -sec).
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec).
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV).
AUGPAR	Augmented data array for input values.
COLAT	Grid colatitude array (radians).
ALOCT	Grid local time array (radians eastward from noon).
FLUX	Flux values for all energy channels at all grid points

4. TEST SPECIFICATION AND EVALUATION

4.1 Test Specification. The set of algorithms encoded as the MSM have been run and tested at Rice University. The results have been studied by comparing the model results with observed data. The version of the program code which was developed at Rice University gives the verified results. To validate the algorithms delivered to the Space Forecast Center, the output from the code run at Rice University will be compared with that run at the Space Forecast Center.

4.2 Test Methods and Constraints.

4.2.1 Test Conditions. The input data will come from Rice University.

4.4 Test Evaluation.

4.4.1 Test Data Criteria. Using the output data furnished from Rice University, the output from the code run at the Space Forecast Center should be the same.

IV. TESTING THE MODEL USING THE CDAW6 EVENT

A. General Description of Event

The Coordinated Data Analysis Workshop #6 (CDAW6) focused its effort on understanding the physical processes which control the flow of energy from the solar wind through the magnetosphere and the eventual release through magnetospheric substorms. The participants studied the first of 2 large substorms which occurred on March 22, 1979. Preceded by a day of quiet conditions both in the solar wind and the magnetosphere, a storm sudden commencement was observed at 0826UT followed by the onset of the expansive phase of the first substorm at 1054UT. The ready availability of a large data set and numerous papers (see Journal of Geophysical Research volume 90, February 1985 for most of them) made it a natural candidate for the preliminary testing of the model's capabilities.

Figure 4.1 is a summary of solar wind plasma parameters at IMP-8 (located slightly to the duskside of the bow shock) and the AE magnetic activity index published as Figure 2 in the overview paper by McPherron and Manka (1985). The major features of these data include:

- 1) An interplanetary shock arrives at IMP-8 at 0821:20UT, as indicated by an increase in solar wind density, velocity and magnetic field strength.
- 2) At 0826UT the shock arrives at the dayside of the earth compressing the magnetosphere causing a sudden storm commencement, which can be seen as a small spike in the AE index.
- 3) The magnetic field turns southward at IMP-8 at 1008UT which begins the stretching of the magnetotail and the growth phase of the first substorm.
- 4) The first substorm expansion onset at 1054UT produced auroral zone magnetic activity (as measured by the AE index) exceeding 1000 nT.

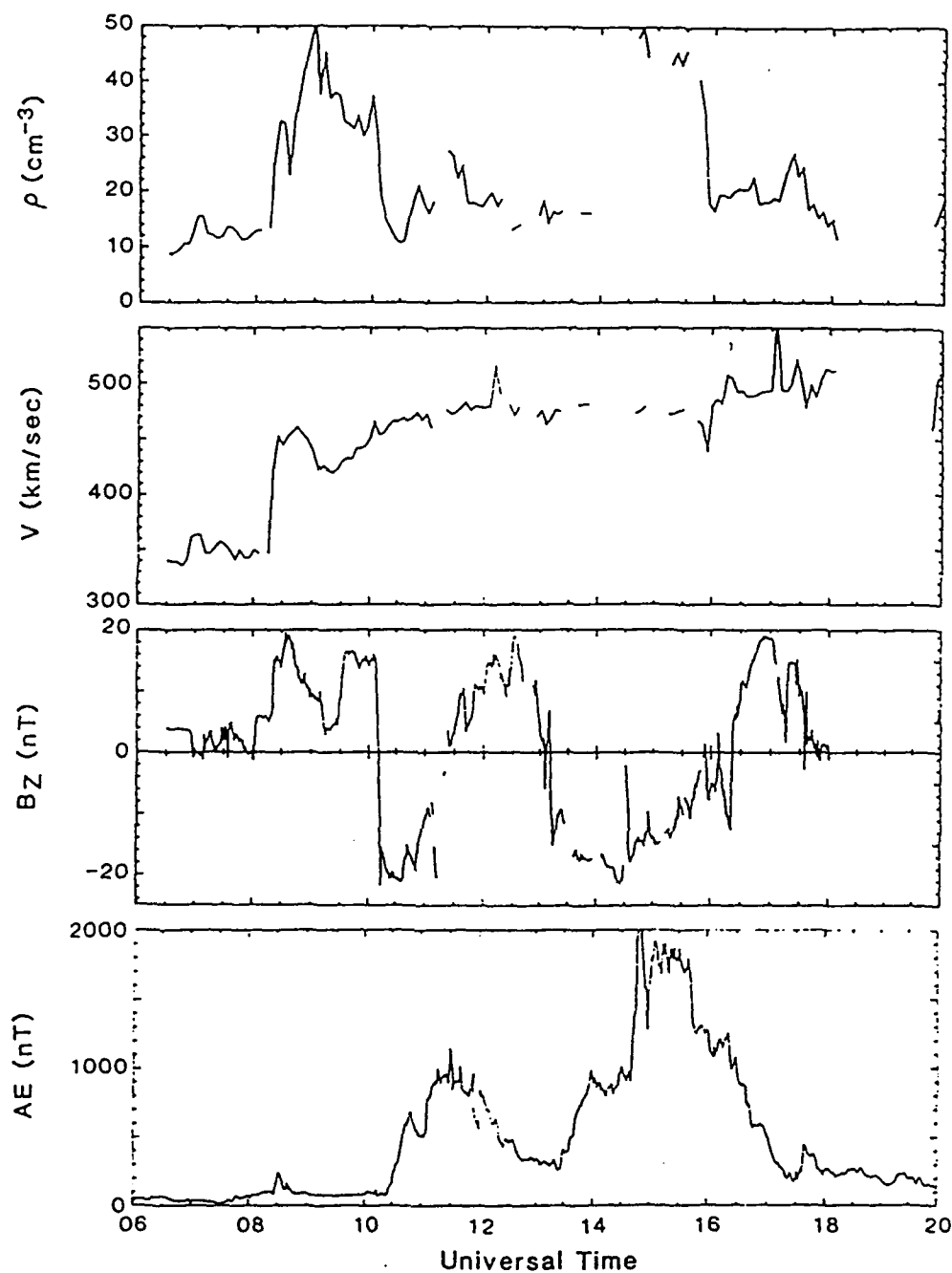


Figure 4.1 Summary of solar wind plasma parameters at IMP 8 on March 22, 1979. Top traces show 5-min averages of solar wind velocity and density. Middle trace shows 1-min values of the GSM Z component of the IMF at IMP 8. Bottom trace presents a 55-station AE index calculated at 1 min time resolution.

5) At 1122UT the IMF becomes northward at IMP-8, magnetic activity starts to decline and the recovery phase of the substorm begins.

B. Model Inputs

1. Magnetic Field Model

After studying the IMF data and Dst index, Dr. G.-H. Voigt subdivided the CDAW-6 interval into 17 time labels (Table 4.1) which represent different configurations of the magnetic field during the event. The magnetic field matrices were then pre-computed for each time. The inputs to the model are

STAND - the magnetopause standoff distance (R_e) from the solar wind data.

Dst - the standard index of ring current intensity (nT).

Aur.Eq.Bd. - the equatorward edge of the auroral zone at midnight (information on how this is done is in Section 2 - Electric Field Model Inputs).

Map(E) - the expected mapping from the Aur.Eq.Bd. (above) to the inner edge of the plasma sheet at local midnight in the equatorial plane of the magnetosphere (R_e). This was estimated using the work by Kivelson et al.(1979), Horowitz et al. (1986), and Gussenhoven et al. (1981) who have calculated K_p -based relationships between the inner edge of the plasma sheet and the equatorward edge of the auroral zone.

From these four observable parameters, Hilmer (1989) has devised a scheme to estimate the remaining input values necessary to run the model:

TILT - Dipole tilt angle (set to 0.0 for the CDAW-6 event).

HJNEAR - Strength of the cross-tail current sheet at the inner edge.

XNEAR - Location of the inner edge of the model plasma sheet at local midnight (R_e).

HJFRAC - Fraction of HJNEAR appearing at and beyond (XNEAR-100) R_e .

Current intensity is constant in the far tail.

LABEL = 1	6:00	UT	Pre-Storm Sudden Commencement
LABEL = 2	8:26	UT	Storm Sudden Commencement
LABEL = 3	10:00	UT	Same as LABEL 2 but little bit more stretch
LABEL = 4	10:20	UT	IMF turns southward
LABEL = 5	10:54 - EPS		Maximum stretch just before onset of first substorm
LABEL = 6	10:54 + EPS		Collapse right after onset of substorm
LABEL = 7	11:30	UT	End of phase "gradually back to normal"
LABEL = 8	12:06	UT	Nominal phase (i.e. nominal model)
LABEL = 9	12:43	UT	Nominal phase
LABEL = 10	13:10	UT	IMF turns southward
LABEL = 11	13:40	UT	Gradual stretching period
LABEL = 12	14:07	UT	Gradual stretching period
LABEL = 13	14:36 - EPS		Maximum stretch just before onset of second substorm
LABEL = 14	14:36 + EPS		Collapse right after onset of substorm
LABEL = 15	15:15	UT	Gradually back to normal after collapse
LABEL = 16	15:50	UT	Gradually back to normal
LABEL = 17	16:00	UT	Beginning of last nominal phase

Table 4.1 Critical time labels for CDAW-6 event.

DY - Scale length for changes in tail structure along the GSM y-axis (R_E). This applies to the westward tail current system.

DFIL - Plasma sheet thickness parameter.

BRN, BRP - Ring-current contributions (nT) at Earth's center from the westward and eastward components, respectively ($BRN < 0$, $BRP > 0$).

RN, RP - Characteristic radii of the westward and eastward ring-current components, respectively.

COLLAP - Fraction of the pre-collapse midnight cross-tail current remaining in the disturbed region (from the inner edge, XNEAR to XNEAR - DELXC). COLLAP = 1.0 means no collapse.

DELXC - The distance from the inner edge (R_E), along the x-axis, which is altered by the addition of eastward tail current. DELXC = 0.0 means no collapse.

DYC - Scale length for changes in tail structure along the GSM y-axis (R_E). Applies to the eastward disruption collapse current. DYC is always less than DY. DYC = 0.0 means no collapse.

Table 4.2 from Dr. R. Hilmer contains a list of these parameters for the CDAW-6 event. Also listed are three output parameters:

MAP(R) - The resulting midnight equatorial mapping distance (R_E).

RC Zero - Characteristic radius for the ring current (R_E). The ring current changes sign at this distance and is an implied particle pressure maximum.

Min-Loc - The minimum $|B|$ and its location in the equatorial plane at midnight.

2. Electric Field Model

The cross polar cap potential was calculated from the IMF data and the empirical formula suggested by Crooker et al. (1982) and presented by Reiff and Luhmann (1986)

$$\phi(\text{kV}) = 22 + 0.069 \sqrt{B} Q1$$

where $Q1 = \alpha(\alpha - \cos \theta) (\alpha^2 + 1 - 2\alpha \cos \theta)^{-1/2}$ for $\cos \theta < \alpha$

	Label	UT	Comment	Stand	Dst	Aur.Eq.Bd.	Map(E)	Map(R)	RC Zero	Min-Loc
1	1	6:00	Pre-SSC	10.58	13.53	66.28	7.01	7.00	3.56	0.72 - 36
2	2	8:26	SSC	7.83	43.20	66.28	7.01	7.01	3.21	0.81 - 26
3	3	10:00	"	7.83	43.20	65.25	6.69	6.69	3.21	1.70 - 20
4	4	10:20	IMF South	8.93	32.56	64.72	6.53	6.52	3.20	1.24 - 20
5	5	10:54	Stretched	8.45	16.61	62.51	5.86	5.83	3.04	0.50 - 17
6	6	10:54+	Collapse	8.45	16.61	62.51			3.04	0.42 - 24
7	7	11:30	Nominal	8.10	-15.30	59.22	4.38	4.33	2.51	3.16 - 21
8	8	12:06	"	8.45	-24.41	58.82	4.24	4.23	2.41	3.21 - 21
9	9	12:43	"	8.77	-18.34	59.77	4.58	4.54	2.60	2.80 - 21
10	10	13:10	IMF South	8.71	-13.78	60.72	4.92	4.94	2.66	2.48 - 20
11	11	13:40		8.64	-9.22	60.73	4.92	4.90	2.64	2.44 - 20
12	12	14:07		8.61	-16.82	60.73	4.92	4.92	2.56	2.11 - 18
13	13	14:36	Stretched	7.58	-22.14	60.13	4.71	4.72	2.42	2.90 - 20
14	14	14:36+	Collapse	7.58	-22.14	60.13			2.42	1.09 - 26
15	15	15:15	Nominal	7.62	-39.61	59.80	4.59	4.58	2.24	2.57 - 22
16	16	15:50	"	7.68	-57.08	59.27	4.40	4.40	2.14	1.49 - 26
17	17	16:00	"	8.50	-60.88	59.14	4.35	4.35	2.14	1.40 - 26

	HJNEAR	XNEAR	HJFRAC	DY	DFIL	COLLAP	DELXC	DYC	BRN	BRP	RN	RP
1	4.3	-7.00	0.25	15.0	2.0	1.0	0.0	0.0	-105.0	117.2	3.2	2.1
2	8.3	-7.01	0.20	12.5	2.0	1.0	0.0	0.0	-50.0	75.9	4.0	1.6
3	12.0	-6.69	0.15	10.0	1.2	1.0	0.0	0.0	-50.0	75.9	4.0	1.6
4	12.7	-6.53	0.15	10.0	1.0	1.0	0.0	0.0	-50.0	72.0	4.0	1.6
5	17.5	-5.86	0.14	7.5	0.8	1.0	0.0	0.0	-230.0	221.6	2.9	2.1
6	17.5	-5.86	0.14	7.5	0.8	0.7	45.0	5.0	-230.0	221.6	2.9	2.1
7	15.0	-4.38	0.15	7.5	1.0	1.0	0.0	0.0	-400.0	354.5	2.7	2.1
8	15.0	-4.24	0.15	7.5	1.0	1.0	0.0	0.0	-490.0	437.1	2.6	2.1
9	15.0	-4.58	0.15	7.5	1.0	1.0	0.0	0.0	-390.0	345.8	2.7	2.1
10	15.0	-4.92	0.15	7.5	1.0	1.0	0.0	0.0	-380.0	340.4	2.7	2.1
11	15.0	-4.92	0.15	7.5	1.0	1.0	0.0	0.0	-540.0	504.9	2.6	2.2
12	17.0	-4.92	0.14	7.5	1.0	1.0	0.0	0.0	-500.0	466.4	2.5	2.1
13	17.0	-4.71	0.14	7.5	1.0	1.0	0.0	0.0	-570.0	522.9	2.5	2.1
14	17.0	-4.71	0.14	7.5	1.0	0.8	35.0	5.0	-570.0	522.9	2.5	2.1
15	15.0	-4.59	0.15	7.5	2.0	1.0	0.0	0.0	-840.0	774.3	2.4	2.1
16	11.0	-4.40	0.20	10.0	2.0	1.0	0.0	0.0	-700.0	615.9	2.5	2.1
17	11.0	-4.35	0.20	10.0	2.0	1.0	0.0	0.0	-680.0	597.9	2.5	2.1

Table 4.2 Input parameters for B-field model for CDAW-6 interval.

$$Q1 = 0$$

$$\text{for } \cos \theta > \alpha$$

$$\alpha = \min \left[\frac{(4B_T B_E)^{1/2}}{60 \text{ nT}}, 1 \right]$$

θ = polar angle of the IMF

B_T = projection of IMF on solar-magnetosphere y-z plane

B_E = magnetic field just inside magnetopause

The polar cap pattern type was also estimated from B_y and B_z IMF data for each time label using the criteria given in chapter 3. For the CDAW-6 interval, the equatorward edge of the diffuse aurora was based on electron data from DMSP, P78, and TIROS satellites.

Table 4.3 contains the input data calculated by Dr. R. A. Wolf and used during the interval for:

VPVC - cross-polar cap potential (kV).

IPATTV - polar cap pattern type.

EDGE - equatorward edge of auroral zone at midnight.

A(L) - radius of ellipse measured in x (sunward) direction.

B(L) - radius of ellipse measured in y (duskward) direction.

DX(L) - sunward displacement of coordinate system center from pole.

DY(L) - duskward displacement of coordinate system center from pole.

3. Boundary Plasma Condition

The invariant temperatures (λ_{kTe} , λ_{kTi}) were estimated from data from the ISEE-2 satellite, as shown in Figure 4.2 and published as Figure 1 in Paschmann et al. (1985). The satellite moved earthward from ~14.8 to ~13.0 Re geocentric distance at ~0200 local time from 1010-1200 UT. However, the published observations do not cover the entire

	1	2	3	4	5	6	7	8
1	Time	VPCV	IPATTV	EDGE	A(1)	A(2)	A(3)	B(1)
2	0 - 0826-e	24	4	66.25	15.47	16.81	21.88	15.67
3	0826+e	24	5	66.25	17.85	19.25	21.88	18.52
4	1000	24	5	65.35	18.75	20.48	22.63	18.63
5	1020	66	3	64.80	18.70	20.66	23.10	17.56
6	1054-e	137	3	62.80	20.88	22.89	24.78	19.05
7	1054+e	137	3	62.80	19.70	21.61	24.78	18.78
8	1130	130	3	59.00	20.94	22.72	27.97	20.12
9	1206	107	3	59.00	20.91	22.71	27.97	19.86
10	1243	88	3	59.90	20.77	22.61	27.21	19.39
11	1310	94	3	60.20	20.72	22.59	26.96	19.31
12	1340	135	3	60.85	20.61	22.47	26.42	19.25
13	1407	161	3	60.60	20.76	22.68	26.63	19.17
14	1436-e	156	3	60.10	21.34	23.15	27.05	20.64
15	1436+e	156	3	60.10	20.74	22.49	27.05	20.49
16	1515	144	2	59.80	20.99	22.71	27.30	20.75
17	1550	127	2	59.25	20.73	22.31	27.76	21.08
18	1600	122	2	59.25	20.33	22.02	27.76	20.19

	9	10	11	12	13	14	15	16
1	B(2)	B(3)	DX(1)	DX(2)	DX(3)	DY(1)	DY(2)	DY(3)
2	14.04	22.83	-1.00	-0.56	-1.87	-0.90	0.00	0.00
3	17.09	22.83	-1.19	-0.81	-1.87	-0.77	0.00	0.00
4	17.40	23.74	-1.54	-1.62	-2.02	-0.64	0.00	0.00
5	16.53	24.30	-1.73	-2.21	-2.10	-0.64	0.00	0.00
6	17.82	26.33	-1.96	-2.48	-2.42	-0.64	0.00	0.00
7	17.50	26.33	-1.36	-1.78	-2.42	-0.64	0.00	0.00
8	18.88	30.18	-1.08	-1.37	-3.03	-0.64	0.00	0.00
9	18.53	30.18	-1.10	-1.40	-3.03	-0.64	0.00	0.00
10	18.04	29.27	-1.27	-1.62	-2.89	-0.64	0.00	0.00
11	17.97	28.96	-1.35	-1.73	-2.84	-0.64	0.00	0.00
12	17.90	28.30	-1.35	-1.73	-2.74	-0.64	0.00	0.00
13	17.79	28.56	-1.56	-1.99	-2.78	-0.64	0.00	0.00
14	19.46	29.06	-1.20	-1.63	-2.86	-0.64	0.00	0.00
15	19.28	29.06	-0.89	-1.15	-2.86	-0.64	0.00	0.00
16	19.49	29.37	-0.68	-0.92	-2.90	-0.64	0.00	0.00
17	20.07	29.92	-0.31	-0.40	-2.99	-0.64	0.00	0.00
18	19.06	29.92	-0.48	-1.62	-2.99	-0.64	0.00	0.00

Table 4.3 Input parameters for E-field model for CDAW-6 interval.

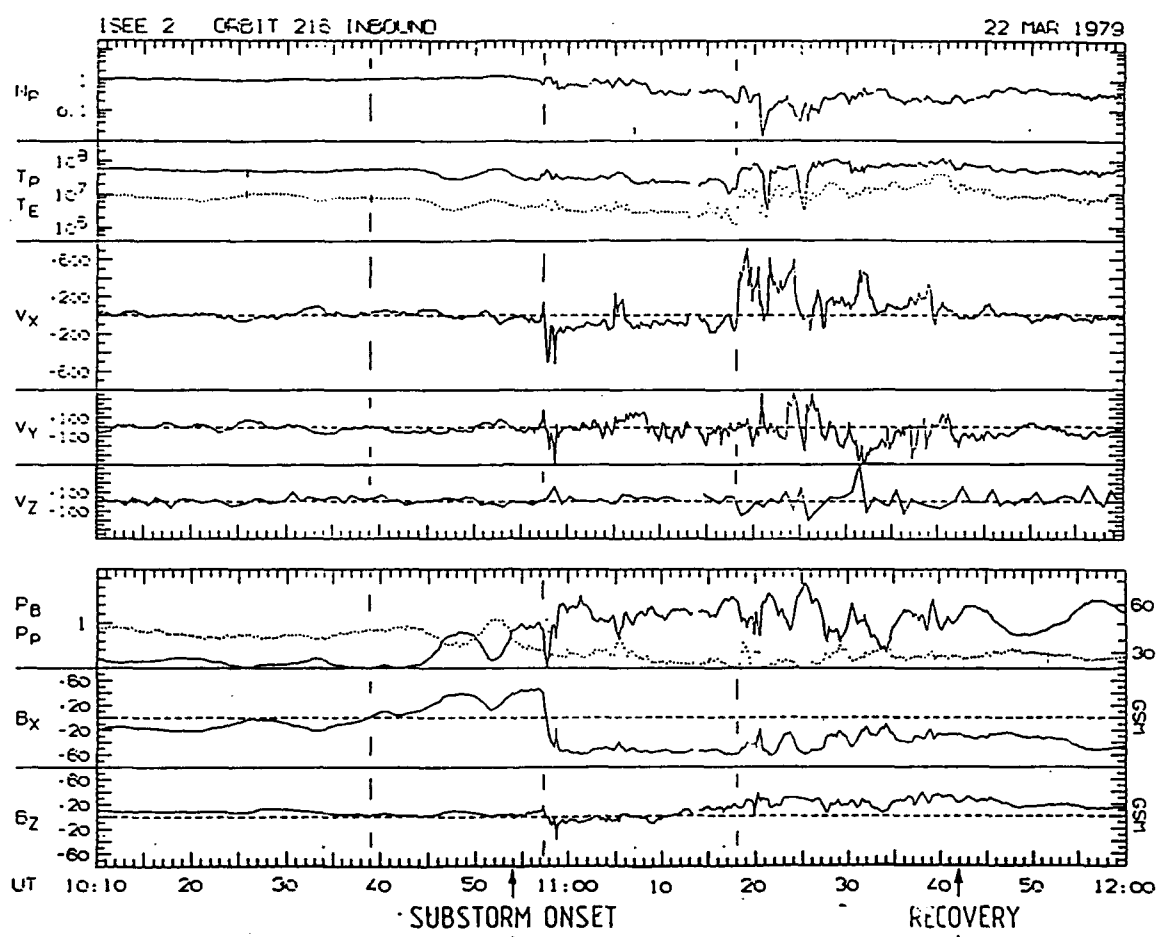


Figure 4.2 ISEE 2 plasma and magnetic field parameters for the time interval 1010-1200 UT on March 22, 1979. T_p and T_e are the ion and electron temperatures (degrees Kelvin).

time interval we want to model, so we have estimated the temperatures from the data and extrapolated to cover the rest of the event. The top of Table 4.4 shows the invariant temperatures calculated for the entire event. The headings are

Label - Time label of magnetic field model (See Table 4.1).

t_{figure} - Time of ISEE-2 data point used to calculate temperature.

r_{figure} - Radial distance of ISEE-2 (Re).

VM_{model} - Flux tube volume at r_{figure} from the magnetic field model.

T_i (°K) - Estimated ion temperature from ISEE-2 data.

T_e (°K) - Estimated electron temperature from ISEE-2 data.

λ_{kTi} - Calculated ion invariant temperature.

λ_{kTe} - Calculated electron invariant temperature.

$\langle |\lambda_i| \rangle$ - average value of ion energy invariant ($= 1.5 \times \lambda_{kTi}$).

$\langle |\lambda_e| \rangle$ - average value of electron energy invariant ($= 1.5 \times \lambda_{kTe}$).

The bottom figure shows the value of η_{tot} calculated from equation 3.24 in Chapter 3 at a distance of 20 Re and local midnight. The headings are

UT_{model} - the universal time of the model.

$B_{\text{lobe}}^2 / 8\pi(\text{model}(20\text{Re}))$ - lobe magnetic pressure at 20 Re and midnight.

$VM_{\text{model}}(20\text{Re})$ - flux tube volume at 20 Re and midnight.

η_{tot} - calculated total invariant density at 20 Re and midnight.

The initial invariant density for each energy channel is given by equation 3.31 in Chapter 3. A list of the energy channels (K), invariant energy (ALAM(K)), and boundary values at 20 Re and local midnight (ETAB(K)) is given in Table 4.5.

4. Initial Plasma Condition

The initial plasma condition is set using $K=32$, the highest electron energy that we model and it corresponds to $\sim 40\text{keV}$ at geosynchronous orbit. Using the 30-45 keV electron channel of the 1977-007 and 1976-059 spacecraft we estimated the average flux

Label	t _{figure}	r _{figure}	VM _{model}	T _i (°K)	T _e (°K)	λ_{kTi}	λ_{kTe}	$\langle \lambda_i \rangle$	$\langle \lambda_e \rangle$
1	10:26am	14.5	1.111	6.4×10^7	1.0×10^7	4960	775	7440	1162.5
2	10:26am	14.5	1.124	6.4×10^7	1.0×10^7	4908	767	7362	1150.5
3	10:26am	14.5	1.147	6.4×10^7	1.0×10^7	4810	752	7215	1128.
4	10:26am	14.5	1.121	6.4×10^7	1.0×10^7	4921	769	7381.5	1153.5
5	10:26am	14.5	1.453	6.4×10^7	1.0×10^7	3797	593	5695.5	889.5
6	10:26am	14.5	1.246	6.4×10^7	1.0×10^7	4428	692	6642	1038
7	11:50am	13.16	1.719	1.0×10^8	2.0×10^7	5015	1003	7522.5	1504.5
8	11:50am	13.16	1.730	1.0×10^8	2.0×10^7	4983	996	7474.5	1494
9	11:50am	13.16	1.649	1.0×10^8	2.0×10^7	5228	1045	7842	1567.5
10	10:26am	14.5	1.287	6.4×10^7	1.0×10^7	4287	670	6430.5	1005
11	10:26am	14.5	1.339	6.4×10^7	1.0×10^7	4120	644	6180	966
12	10:26am	14.5	1.316	6.4×10^7	1.0×10^7	4192	655	6288	982.5
13	10:26am	14.5	1.346	6.4×10^7	1.0×10^7	4099	640	6148.5	960
14	10:26am	14.5	1.296	6.4×10^7	1.0×10^7	4257	665	6385.5	997.5
15	11:50am	13.16	1.593	1.0×10^8	2.0×10^7	5412	1082	8118	1623
16	11:50am	13.16	1.642	1.0×10^8	2.0×10^7	5250	1050	7875	1575
17	11:50am	13.16	1.613	1.0×10^8	2.0×10^7	5345	1069	8017.5	1603.5

Label	UT _{model}	$B_{lobe}^2/8\pi(\text{model } (20R_E))$	VM _{model} (20R _E)	η_{tot}
1	0 - 0826-ε	8.049×10^{-10}	0.6049	1.966×10^{21}
2	0826+ε	20.019×10^{-10}	0.6050	4.940×10^{21}
3	1000	33.054×10^{-10}	0.7366	5.088×10^{21}
4	1020	35.461×10^{-10}	0.7335	5.393×10^{21}
5	1054-ε	65.914×10^{-10}	0.9300	7.178×10^{21}
6	1054+ε	34.808×10^{-10}	0.6462	8.075×10^{21}
7	1130	47.003×10^{-10}	0.8949	4.111×10^{21}
8	1206	49.956×10^{-10}	0.9148	3.912×10^{21}
9	1243	46.720×10^{-10}	0.8730	4.170×10^{21}
10	1310	46.792×10^{-10}	0.8709	5.137×10^{21}
11	1340	46.601×10^{-10}	0.8631	5.636×10^{21}
12	1407	55.492×10^{-10}	0.8923	6.069×10^{21}
13	1436-ε	56.765×10^{-10}	0.9280	5.757×10^{21}
14	1436+ε	43.138×10^{-10}	0.8166	5.799×10^{21}
15	1515	45.457×10^{-10}	0.7328	6.071×10^{21}
16	1550	31.912×10^{-10}	0.6988	4.948×10^{21}
17	1600	31.526×10^{-10}	0.6860	5.028×10^{21}

Table 4.4 Invariant temperature and total invariant density calculated from CDAW-6 interval.

<u>K</u>	<u>ALAM(K)</u>	<u>ETAB(K)</u>
1	374.80640	0.86136E+20
2	1202.52612	0.15263E+21
3	2146.50952	0.19207E+21
4	3223.09546	0.21585E+21
5	4450.89844	0.22572E+21
6	5851.17187	0.22246E+21
7	7448.13672	0.20738E+21
8	9269.42187	0.18272E+21
9	11346.53120	0.15164E+21
10	13715.41020	0.11792E+21
11	16417.03520	0.85332E+20
12	19498.16410	0.56979E+20
13	23012.07810	0.34758E+20
14	27019.57030	0.19146E+20
15	31589.98440	0.93944E+19
16	36802.38670	0.62709E+19
17	-58.56357	0.86136E+20
18	-187.89493	0.15263E+21
19	-335.39233	0.19207E+21
20	-503.60913	0.21585E+21
21	-695.45410	0.22572E+21
22	-914.24683	0.22246E+21
23	-1163.77295	0.20738E+21
24	-1448.34888	0.18272E+21
25	-1772.89771	0.15164E+21
26	-2143.03564	0.11792E+21
27	-2565.16455	0.85332E+20
28	-3046.59180	0.56979E+20
29	-3595.64160	0.34758E+20
30	-4221.81250	0.19146E+20
31	4935.93750	0.93945E+19
32	-5750.37891	0.62709E+19

Table 4.5 A list of the energy channels (K), invariant energy (ALAM(K)), and boundary values at 20 Re and local midnight (ETAB(K)).

to be $10^{1.48} \text{ cm}^{-2}\text{s}^{-1}\text{sr}^{-1}\text{eV}^{-1}$ (See Figure 4.3). The other parameters needed to solve equation 3.40 are:

ALAM(K) - energy invariant for energy K=32; $= -5750.4 (\text{Re}/nT)^{2/3}$

VM - magnetic flux tube volume at 6.6 Re on dawn/dusk meridian

$$= 6.82 (\text{Re}/nT)^{2/3}$$

$\lambda_{k\max}$ - defined in equation 3.33a in Chapter 3

$$= -7451.8 \text{ eV} (\text{Re}/nT)^{2/3}$$

$\lambda_{k\min}$ - defined in equation 3.32b in Chapter 3

$$= -5343.2 (\text{Re}/nT)^{2/3}$$

Solving for EETA(32), the invariant density at geosynchronous orbit is

$$\text{EETA}(32) = 1.666\text{E}17$$

The invariant density for K=32 at the back boundary is given in Table 4.4 as 0.6271E19.

The initial back boundary at local midnight at 15.87 Re. Solving equation 3.42 in Chapter 3 to get the power law coefficient:

$$\frac{\text{EETA}(\text{Geosynchronous})}{\text{EETA}(\text{Boundary})} = \frac{1.666\text{E}17}{0.627\text{E}19} = 0.0266 = \left[\frac{6.6 \text{ Re}}{15.87 \text{ Re}} \right]^p$$

$$p = 4.134$$

Previous runs have shown the 'Kivelson' effect is not important for 40 keV electrons at local midnight, and it is neglected for adjusting the η values. Solving equation 3.44 in Chapter 3, the revised density becomes

$$\eta_{\text{tot}}(r) = (1.966\text{E}21) \left[\frac{r}{15.87} \right]^{4.134}$$

The initial density values for entire grid are calculated using

$$\text{EETA}_{\text{initial}}(K, I, J) = \left[\frac{\eta_p(\text{midnight})}{1.966\text{E}21} \right] \left[\frac{r(I, J)}{15.87} \right]^{4.134}$$

where $\eta_p(\text{midnight})$ is the initial value for energy species K at midnight.

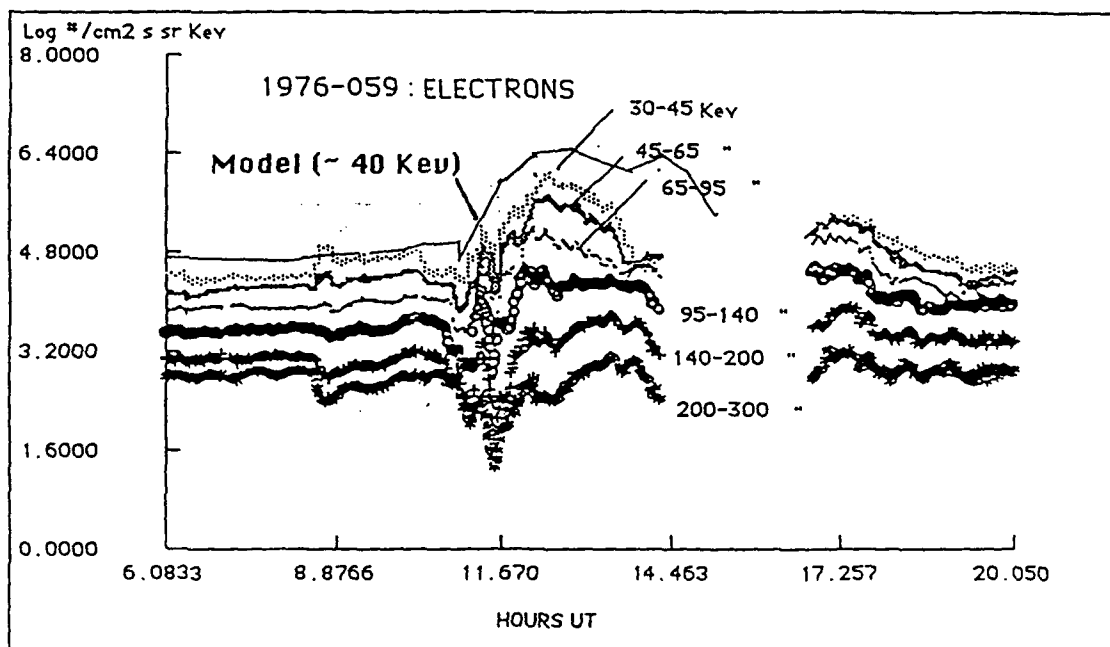
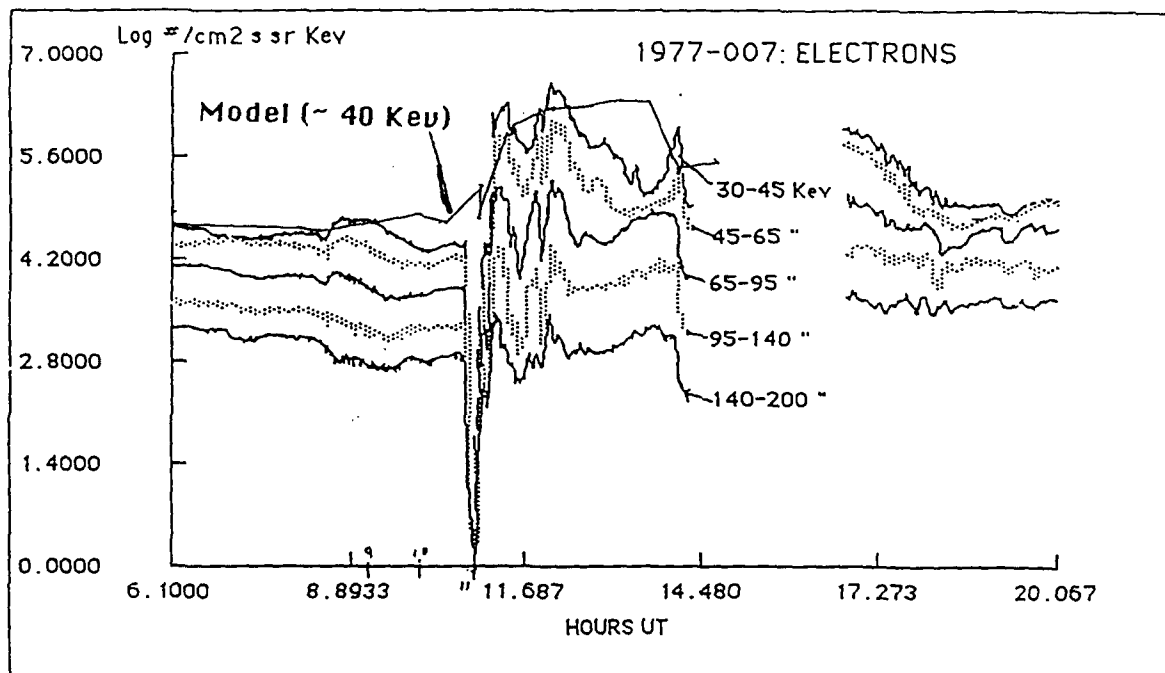


Figure 4.3 Calculated 40 keV geosynchronous electron flux values and 1977-077 and 1976-059 observed 40 keV electron flux values.

The plasma sheet depletion factor (f_{FP}) defined in equation 3.35 of Chapter 3 was set to 5 and was activated at the substorm onset at 10:54:01 UT.

C. Model Results

Figure 4.4 shows the time evolution of the invariant density (η) in the equatorial plane for our highest energy electrons (~ 40 keV at synchronous orbit). Figure 4.5 is a similar plot for "medium energy" ions (~ 40 keV at synchronous orbit); and Figure 4.6 represents low energy ions (~ 2.5 keV at synchronous orbit). The contours represent contours of constant η ; that is, constant numbers of particles per unit magnetic flux. The contours differ by a factor of $10^{0.2}$ in density level. In the plots, the sun is to the left. The outermost contour represents approximately the outer boundary of our calculations. Some physical features include

- 1) Collapse of the magnetic field causes the inner edge of the plasma sheet near local midnight to move earthward approximately 1 Re at the onset of the expansion phase of the first substorm at 1054UT.
- 2) Electrons drift eastward after the injection, with high densities reaching local noon at approximately 1200UT.
- 3) Medium energy ions drift westward from the midnight region, leading to high densities at local noon beginning about 1200UT.
- 4) Subsequently the afternoon inner edge moves gradually earthward.

The output from the model was compared with data available from the CDAW-6 interval, which included

- 1) Available data from geosynchronous orbit
 - a) 1-16 keV ions in the afternoon sector just after noon UT (GEOS-2)
 - b) 40 keV electrons around 0200 LT at 1100UT (1977-077)
 - c) 40 keV electrons around dawn at 1100UT (1976-056)
- 2) Low orbit data

$\lambda = 5750$ electrons (~ 40 keV at synchronous orbit)

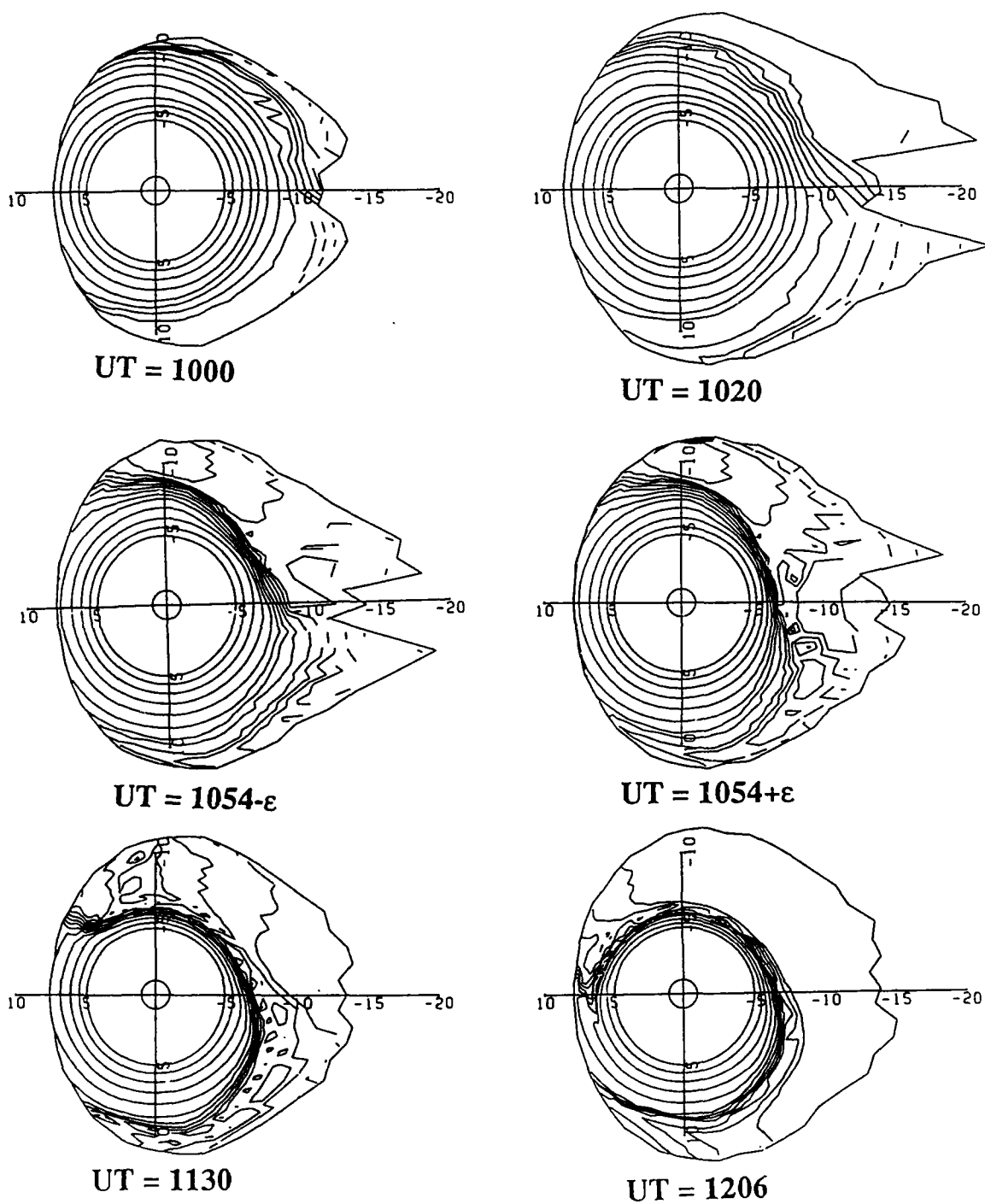


Figure 4.4 Time evolution of invariant density (η) in the uatorial plan for highest energy electrons (~ 40 keV at geosynchronous).

$\lambda = 5851$ ions (~ 40 keV at synchronous orbit)

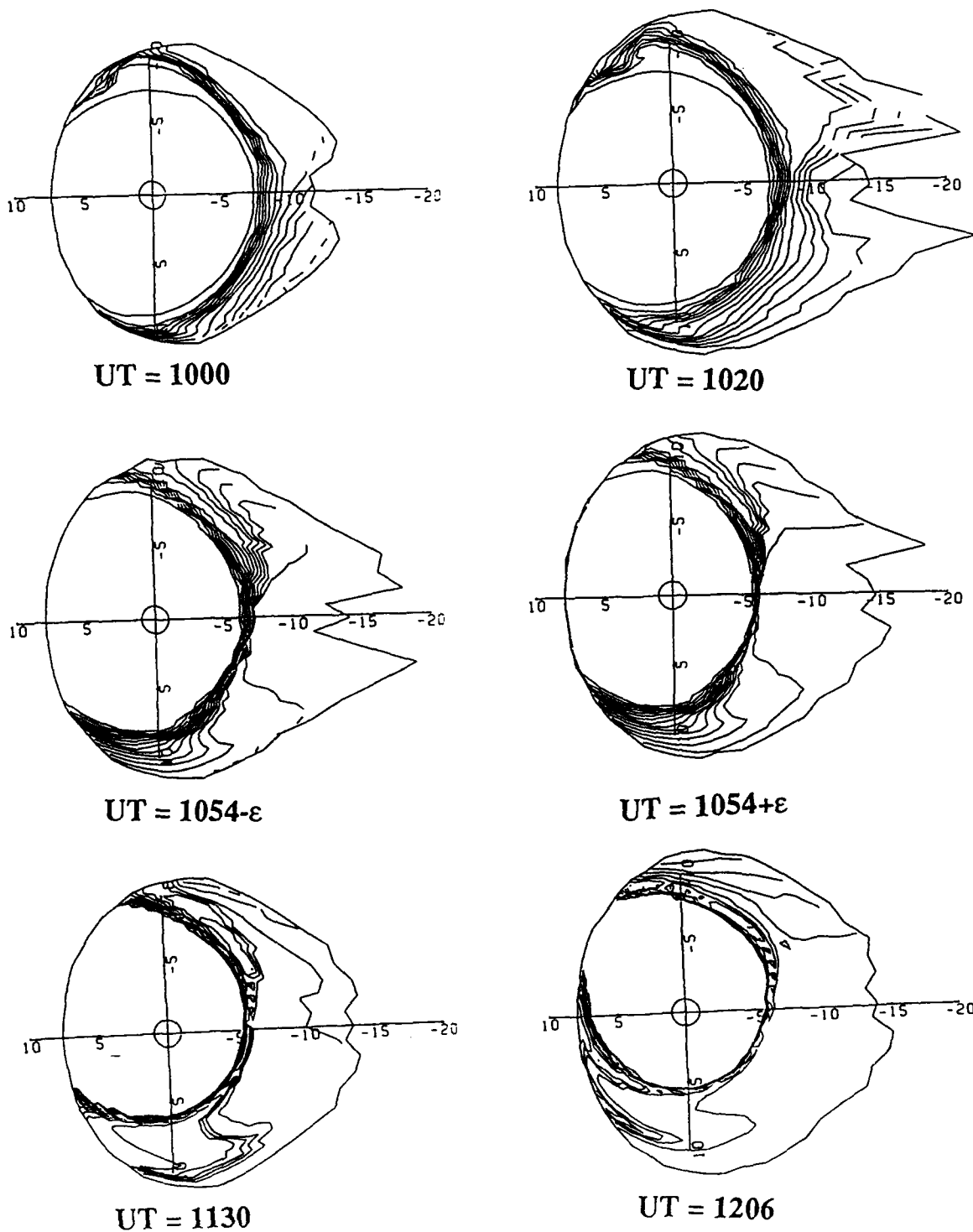


Figure 4.5 Time evolution of invariant density (η) in the equatorial plan for medium energy ions (~ 40 keV at geosynchronous).

$\lambda = 375$ ions (~ 2.5 keV at synchronous orbit)

55

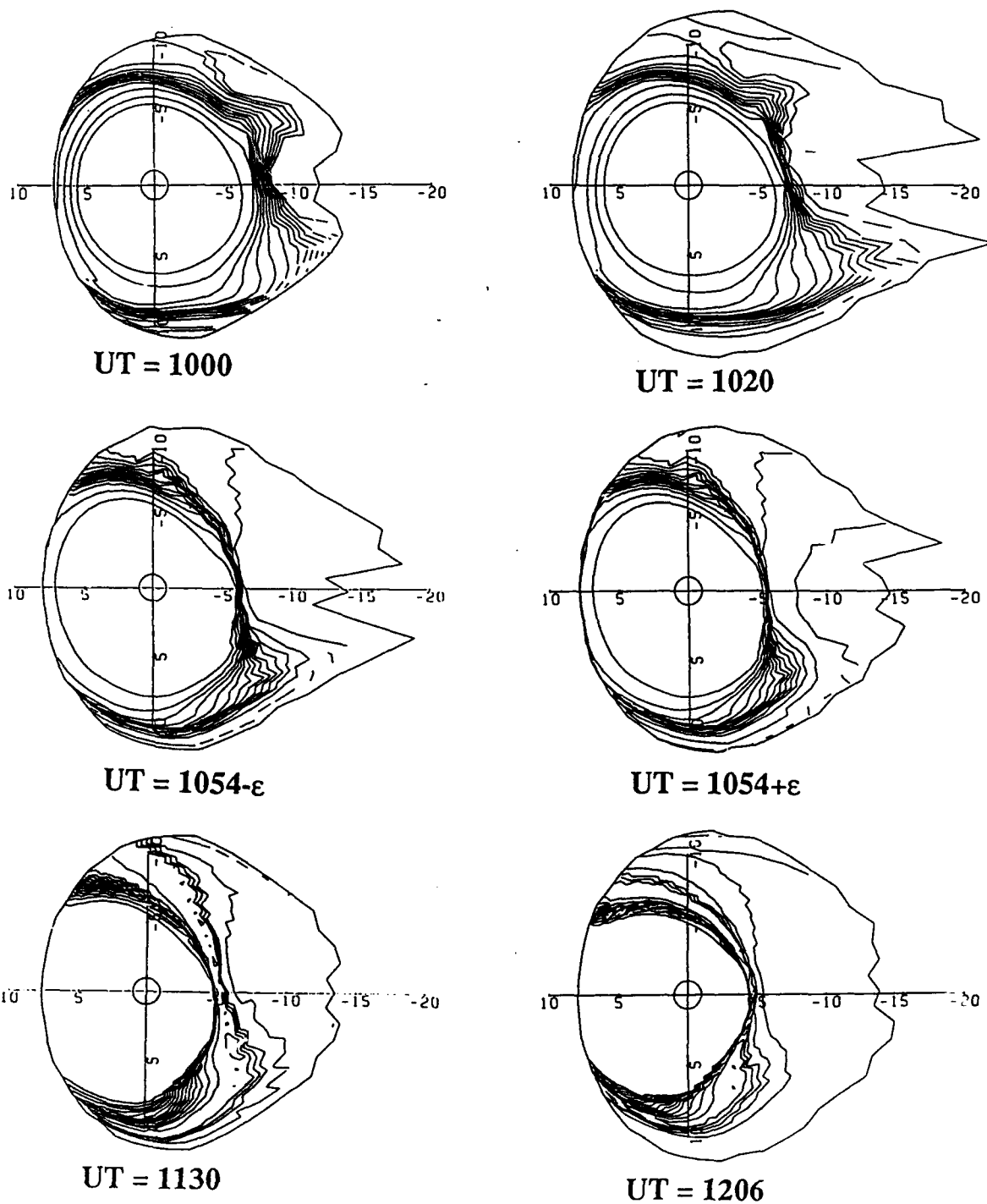


Figure 4.6 Time evolution of invariant density (η) in the equatorial plan for low energy ions (~ 2.5 keV at geosynchronous).

a) Precipitating electron flux from DMSP

Figure 4.3 shows the electron flux measured at the geosynchronous satellites and compares them with the highest electron energy channel ($K=32$) in the model which corresponds to approximately 40 keV at geosynchronous orbit. The arrival times of approximately 1100UT for 1977-077 and 1130UT for 1976-059 show good agreement. However, in both cases the fluxes are too high by a factor of 2.

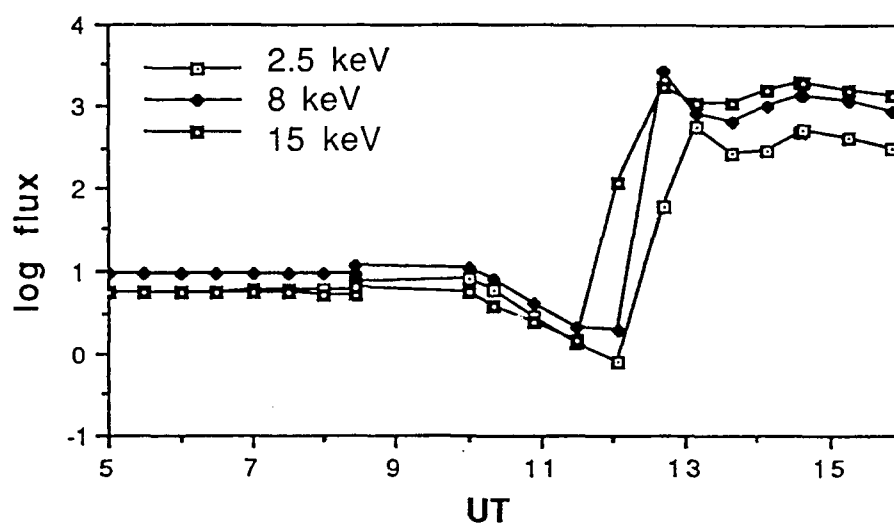
The ion energy channels for $K=1, 2$, and 3 in the model correspond approximately to geosynchronous energies of 2.5 keV, 8.0 keV, and 15.0 keV respectively. A comparison of the model ion flux with the GEOS-2 data is shown on Figure 4.7. Again the arrival times and energy dispersion are in good agreement with the 15 keV ions arriving around 1200UT, 8.0 keV ions arriving around 1230UT, and the 2.5 keV ions arriving about 1330UT. The peak fluxes show the same general tendency to be high by about a factor of 2.

The graphs of the DMSP precipitating electron energy flux are given as Figures 4.8 and 4.9. The model values normalized to the data are represented by overlays. The model does not calculate precipitation in the region poleward of our main modeling region; that excluded region includes the polar cap and the higher-latitude part of the auroral zone so this is not included. The data shown in Figure 4.8 are for the pre-substorm period. In all three examples both the shape and the magnitude of the fluxes agree well with the satellite data. Figure 4.9 presents the results after the substorm onset and Figure 4.10 is the substorm recovery period. As with the geosynchronous fluxes, for both cases the magnitude is again too high by a factor of 2.

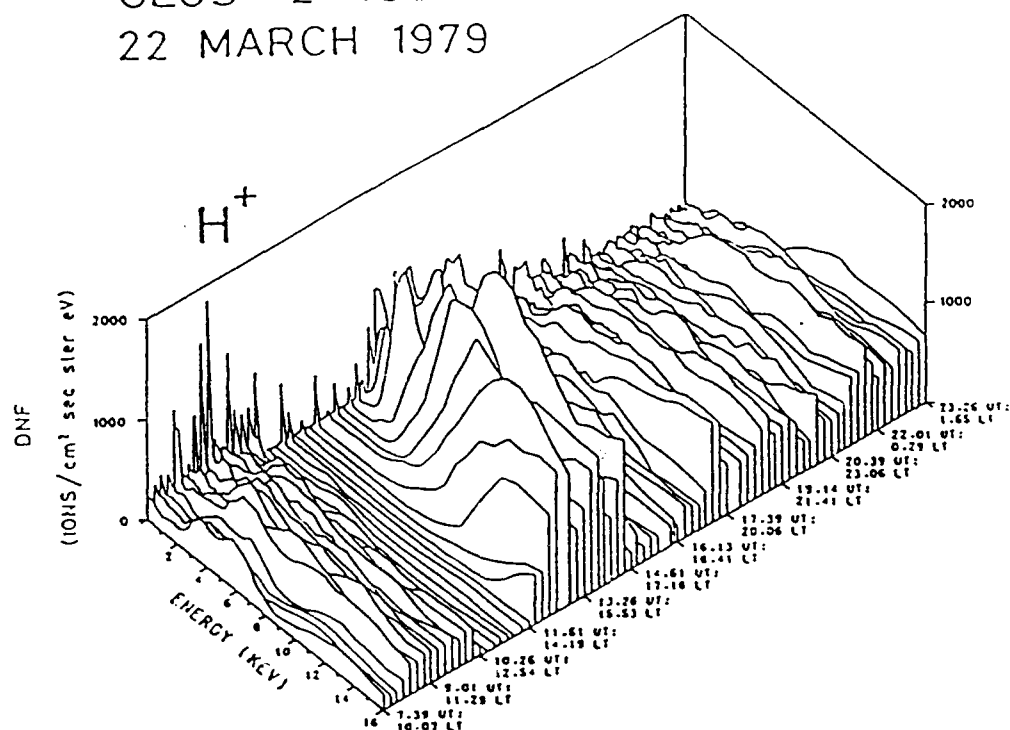
D. Interpretation and Implication

This model represents an attempted simulation of the inner-magnetosphere processes associated with magnetospheric substorms, including the magnetic field collapse at the onset of the substorm expansion phase and the subsequent injection of particles into

Ion Flux at GEOS 2



GEOS 2 ICE
22 MARCH 1979



After Stokholm, Amata, Balsiger, Candidi, Orsini, and Anderson, J.
Geophys. Res., 90, pp 1253-1261, 1985

Figure 4.7 Calculated ion flux values and GOES 2 observed ion flux values.

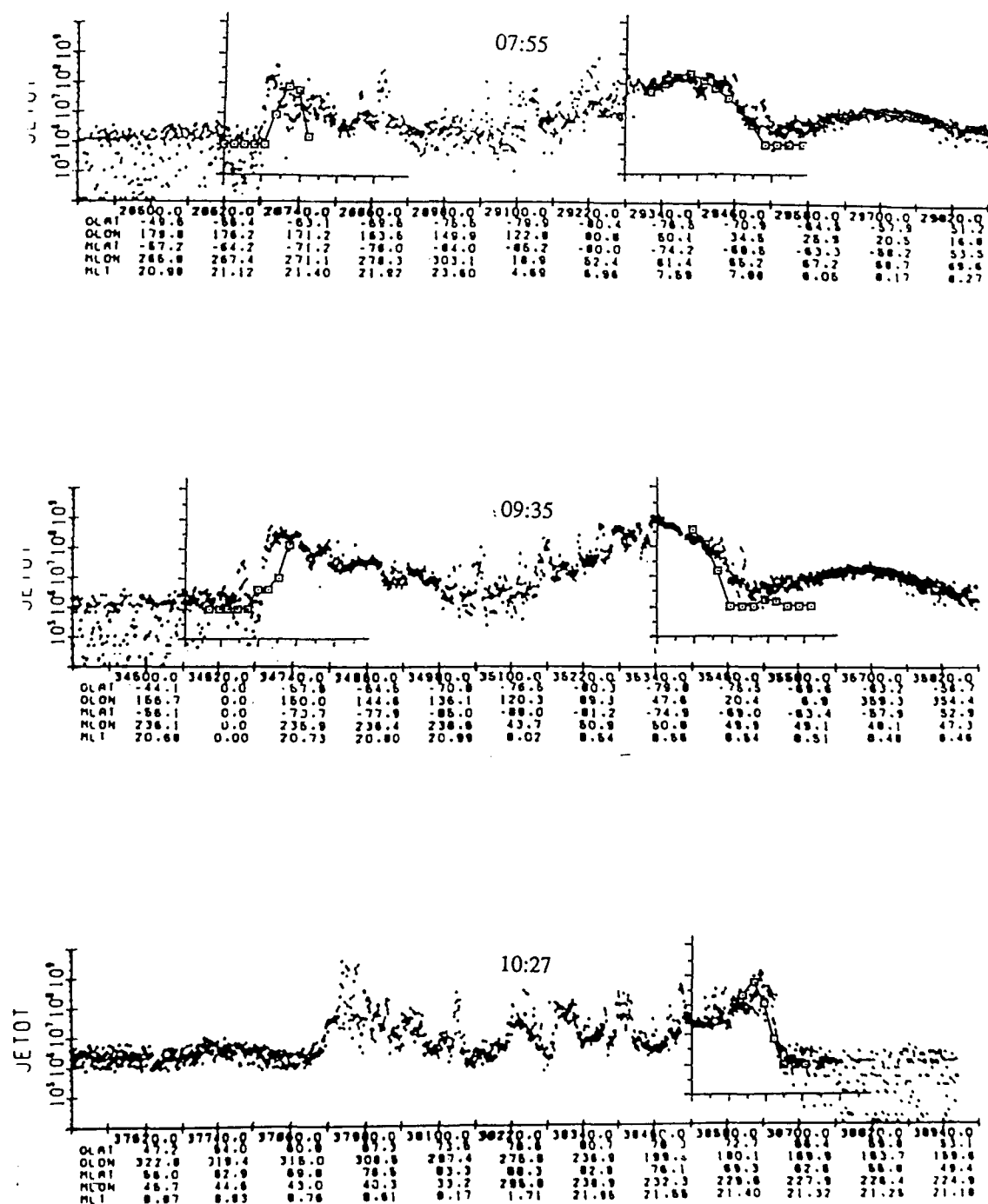


Figure 4.8 DMSP precipitating electron energy flux before substorm onset.

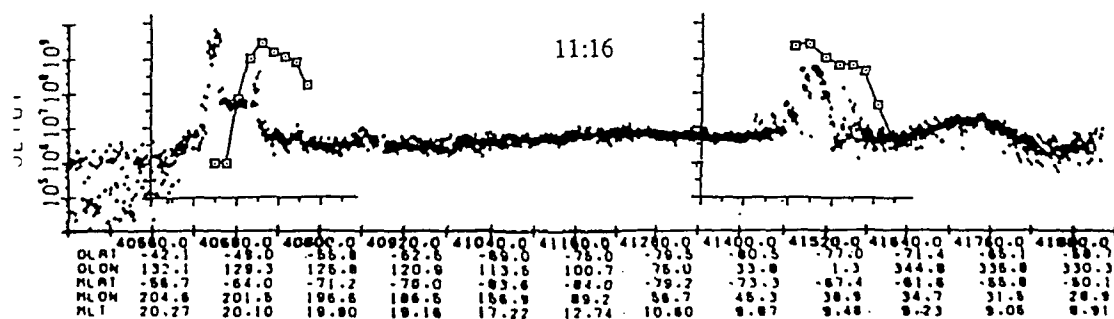


Figure 4.9 DMSP precipitating electron energy flux after substorm onset.

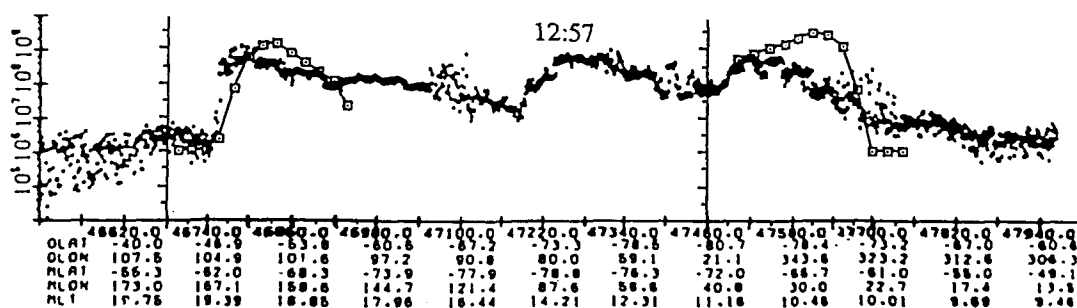
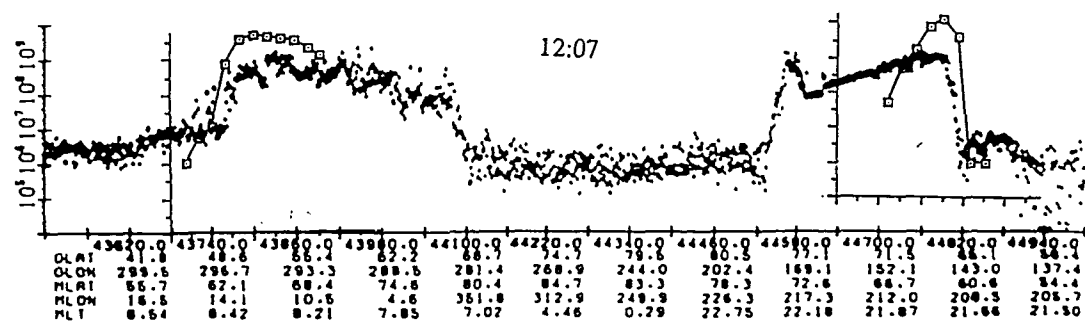


Figure 4.10 DMSP precipitating electron energy flux during substorm recovery period.

geosynchronous orbit. The arrival times of the electrons near dawn and the ions which have drifted to noon local time agree with the satellite data. However, assuming that middle plasma sheet flux tubes are convected adiabatically through the inner plasma sheet to synchronous orbit in a substorm with allowances for precipitation loss, the model results imply unrealistically high synchronous fluxes and unrealistically high precipitation rates. In addition, the modeled, geosynchronous 40 keV electron flux does not decrease during the recovery phase late in the event as the data shows. It may be that we are underestimating the pitch-angle scattering rate for these particles. The Hones (1979) substorm theory implies that a portion of the plasma on tail-plasma-sheet flux tubes escapes down the tail as a plasmoid. Therefore, for the initial model runs, we decreased the plasma sheet density by a factor of 2 at substorm onset. This still produced fluxes and precipitation too high by a factor of 5. The results presented here, from a later model run, assume a factor-of-5 depletion after substorm onset. The model predictions and observations agree to within a factor of 2 in flux levels and precipitation. The implication is that the substorm process must involve a powerful electron loss process in addition to precipitation, or the flux tubes which convect into geosynchronous orbit have previously been depleted.

TEST ANALYSIS REPORT

TEST ANALYSIS REPORT - 1

Test Analysis Report

2. TEST ANALYSIS

2.1 Test The set of algorithms encoded as the MSM have been run and tested at Rice University. The results have been studied by comparing the model results with observed data. The version of the program code which was developed at Rice University gives the verified results. To validate the algorithms delivered to the Space Forecast Center, the output from the code run at Rice University will be compared with that run at the Space Forecast Center.

2.1.3 Performance. Using the output data furnished from Rice University, the output from the code run at the Space Forecast Center should be the same.

3. SUMMARY AND CONCLUSIONS

3.1 Demonstrated Capability. This test verifies that the algorithms developed for the Air Weather Service at Rice University are running properly at the Space Forecast Center.

DATABASE SPECIFICATION

DATABASE SPECIFICATION - 1

Database Specification

2.1 Database Identification.

AWS Environmental Database - Presently all the input data are on disk files, some of which will no longer be needed when the Environmental Data Base is operational. Attached is a listing of the input parameters needed to run the MSM and a proposed data transfer method.

MSM Database - Below is a list of the input files needed to restart the model.

BOxxxxxx	The magnetic field matrices needed for the model run.
COORD	File of values to set up the coordinate system.
DKTABLE	Decay times for computing charge-exchange loss of ions
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model. (km/sec)
MSMIN	Input start and stop times, start direct-access record number, and sunspot number

Following is a list of files output from the MSM.

V	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
VNORTH	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
VSOUTH	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
VM	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
BMIN	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
XMIN	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)

DATABASE SPECIFICATION - 2

YMIN	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
ZMIN	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
BNDLOC	Output volume - number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
EFLUX	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times number of energy channels \times 4 (bytes)
FLXSUM	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 3 particle species \times 4 (bytes)
EAVG	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 3 particle species \times 4 (bytes)
AUGPAR	Output volume - number of data elements \times number of time steps \times 4 (bytes)
COLAT	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
ALOCT	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
FLUX	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times number of energy channels \times 4 (bytes)

2.3.1 Conceptual Model. The MSM will interrogate the Environmental Data Base at mark times for all physical parameters needed by the model. Attached is a copy of the MSM Control and Data Flow Diagram. Subroutine PARGEN within the MSM code will interpolate or extrapolate as necessary to adjust the data to the model time. Subroutine FECON then calculates the parameters needed within the program. The model is capable of running with no data except Kp. If Kp is missing, the program so states and stops.

Data from previous MSM runs is communicated through the direct-access record number read in file MSMIN. It specifies at which time label of the previous run to begin the current run. Zero means the start of a new run.

3.3.2 Content.

BOxxxxxx	The magnetic field matrices needed for the model run.
COORD	File of values to set up the coordinate system.

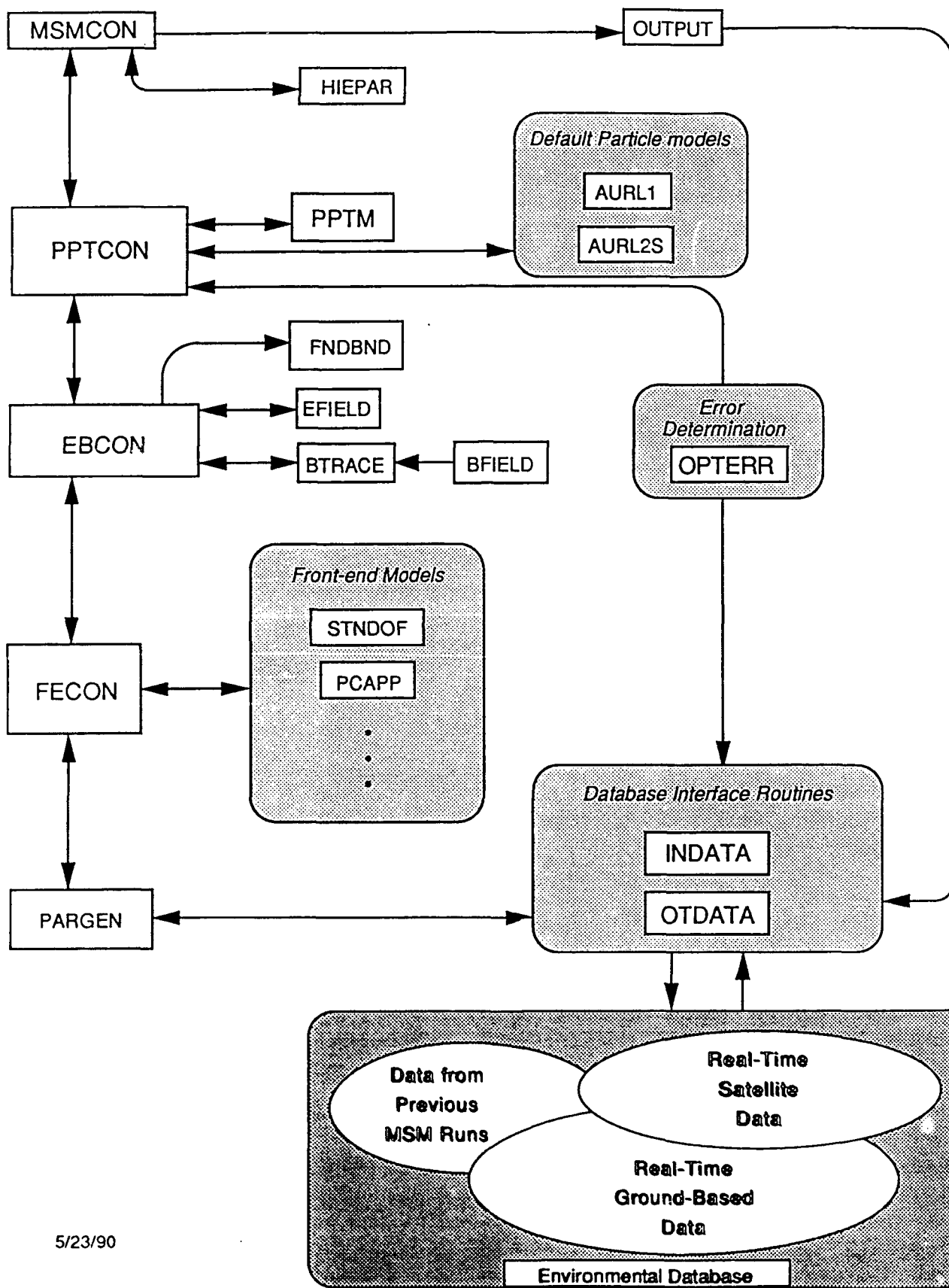
DATABASE SPECIFICATION - 3

EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate.
HARDY	Coefficients used by the Hardy precipitating electron model.
IONENG	Coefficients used as input for the ion precipitating model.
IONNUM	Coefficients used as input for the ion precipitating model. (km/sec)

Following is a list of files output from the MSM.

V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts).
VNORTH	Northern hemisphere electric potential distribution (Volts).
VSOUTH	Southern hemisphere electric potential distribution (Volts).
VM	Flux tube volume $((R_e/nT)^{-2/3})$.
BMIN	Equatorial magnetic field strength (nT).
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).
BNDLOC	Location of outer boundary of detailed particle traces.
EFLUX	Precipitating energy flux array (ergs/cm ² -sec).
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec).
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV).
AUGPAR	Augmented data array for input values.
COLAT	Grid colatitude array (radians).
ALOC	Grid local time array (radians eastward from noon).
FLUX	Flux values for all energy channels at all grid points

MSM Control & Data Flow Diagram



5/23/90

PARAMETER	NAME	UNITS	IO	Time Tag	Geo. Latitude	Geo. Longitude	Altitude	Mag. Local Time
-----------	------	-------	----	----------	---------------	----------------	----------	-----------------

INPUT ONLY:

Kp	FKP	NONE	I	X				
Dst	DST	NANOTESLA	I	X				
C2	C2	NONE	I	X				
LOW LAT AURORAL BOUNDARY	DLATAZ	DEGREES	I	X			X	X
EQ. EDGE R1 CURRENTS	EDGER1	DEGREES	I	X			X	X
EQ. EDGE R2 CURRENTS	EDGER2	DEGREES	I	X			X	X
POLAR CAP POTENTIAL	PCP	KILOVOLTS	I	X				
POLAR CAP POTENTIAL PATTERN	XIPATT	NONE	I	X				
POLAR CAP BOUNDARY LATITUDE	PCBND	T.B.D.	I	X				
PRECIPITATION POWER INDEX	P	NONE	I	X				
GEO MAG X COMP.	GEOMGX	NANOTESLA	I	X				X
GEO MAG Y COMP.	GEOMGY	NANOTESLA	I	X				X
GEO MAG Z COMP.	GEOMGZ	NANOTESLA	I	X				X
GEO ELECTRON FLUX - CHANNEL(1)	EFLX(1)	ELEC/CM ² .s	I	X				X
GEO ION FLUX - CHANNEL(1)	GFLX(1)	IONS/CM ² .s	I	X				X
SOLAR WIND VELOCITY	SWVEL	KM/s	I	X	X	X	X	X
SOLAR WIND DENSITY	SWDEN	PROT/CM ³	I	X	X	X	X	X
IMF X COMPONENT	XIMFBX	NANOTESLA	I	X	X	X	X	X
IMF Y COMPONENT	XIMFBY	NANOTESLA	I	X	X	X	X	X
IMF Z COMPONENT	XIMFBZ	NANOTESLA	I	X	X	X	X	X

Data Transfer Method

1. MSM will access the environmental database input data files by calling a subroutine, **INDATA**.
2. The call will be of the form:
CALL INDATA(parm-name, STARTT, ENDT, NDIM, DARRY, NUMNUM)
 Where:
parm-name is a valid parameter name from the MSM input parameter list.
STARTT is the starting time for this data request.
ENDT is the ending time for this data request.
NDIM is the horizontal dimension of **DARRY** and the maximum number of data points that may be returned.
DARRY is a two dimensional array to be retrieved from the database. The form of **DARRY** is shown on the next page. The size of **DARRY** will be **NDIM** by 7.
NUMNUM is the number of data values placed in **DARRY** by **INDATA**.
3. The system controller will periodically update the input data files.
4. Rice does not plan to write **INDATA**.
5. All parameters will be in single precision, floating point format.
6. Output data will be passed from MSM to the environmental database via a subroutine, **OTDATA**.
7. The call to **OTDATA** will be defined similarly to **INDATA**.
8. Rice does not plan to write **OTDATA**.

Format for DARRY

NDIM

→

7	Data value at t1	Data value at t2	Data value at t3	Etc.
↓	Time tag at t1	Time tag at t2	Time tag at t3	
	Latitude	Latitude	Latitude	
	Longitude	Longitude	Longitude	
	Altitude	Altitude	Altitude	
	mag. local time	mag. local time	mag. local time	

IMPLEMENTATION PROCEDURES

IMPLEMENTATION PROCEDURES - 1

Implementation Procedures

3.1.5 Detailed Procedures. Following are the files needed to run the MSM, including those to be replaced by calls to the environmental data base. Those required for a restart are so labelled.

BOxxxxxx	The magnetic field matrices needed for the model run. Required for restart.
CLAPSE	File containing values giving the collapse/no collapse state of the magnetosphere. To be replaced by call to AWS environmental data base, when complete.
COORD	File of values to set up the coordinate system. Required for restart.
DST	Dst values for the event. To be replaced by call to AWS environmental data base, when complete (nT).
DKTABLE	Decay times for computing charge-exchange loss of ions
EFCOEF	Coefficients from Heppner-Maynard model which are input to the electric field model. Required for restart.
ENCHAN	Input values for the number of energy channels, particle type and energy channels for the model to simulate. Required for restart.
EQEDGE	Values of the equatorward edge of the auroral oval at midnight used as input into the electric field model. To be replaced by call to AWS environmental data base, when complete (degrees).
FKP	Kp data for the event. To be replaced by call to AWS environmental data base, when complete. Required for restart.
HARDY	Coefficients used by the Hardy precipitating electron model. Required for restart.
IONENG	Coefficients used as input for the ion precipitating model. Required for restart.
IONNUM	Coefficients used as input for the ion precipitating model. Required for restart.
MSMIN	Input start and stop times, start direct-access record number, and sunspot number
PCP	Cross-polar cap potential values for event used as input to the electric field model. To be replaced by call to AWS environmental data base, when complete (kV).
SWDEN	Solar wind density values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (particles/cm ³)
SWVEL	Solar wind velocity values used to calculate the standoff distance for the event. To be replaced by call to AWS environmental data base, when complete. (km/sec)
XIPATT	Polar cap pattern type used as input to electric field model for the event. To be replaced by call to AWS environmental data base, when complete.
V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts).
	Output volume - number of latitudinal gridpoints × number of longitudinal gridpoints × number of time steps × 4 (bytes)
VNORTH	Northern hemisphere electric potential distribution (Volts).

IMPLEMENTATION PROCEDURES - 2

	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
VSOUTH	Southern hemisphere electric potential distribution (Volts).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
VM	Flux tube volume $((R_e/nT)^{-2/3})$.
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
BMIN	Equatorial magnetic field strength (nT).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (R_e).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
BNDLOC	Location of outer boundary of detailed particle traces.
	Output volume - number of longitudinal gridpoints \times number of time steps \times 4 (bytes)
EFLUX	Precipitating energy flux array (ergs/cm ² -sec).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times number of energy channels \times 4 (bytes)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 3 particle species \times 4 (bytes)
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV).
	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 3 particle species \times 4 (bytes)
AUGPAR	Augmented data array for input values.
	Output volume - number of data elements \times number of time steps \times 4 (bytes)
COLAT	Grid colatitude array (radians).

IMPLEMENTATION PROCEDURES - 3

ALOCT	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes) Grid local time array (radians eastward from noon).
FLUX	Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times 4 (bytes) Flux values for all energy channels at all grid points Output volume - number of latitudinal gridpoints \times number of longitudinal gridpoints \times number of time steps \times number of energy channels \times 4 (bytes)

4.1.2 Detailed Procedures. To run the MSM:

Start and stop times are specified to the program through the 5 line MSMIN file. Following is an example of MSMIN for the fresh startup of the program.

1988 112 64800	'start year, start day, start time (seconds)'
1988 112 66600	'end year, end day, end time (seconds)'
0	'direct-access record number 0 means new start'
'30 minute test run'	'up to 80 character run identification'
0.0	'sunspot number=0.'

The MSMIN configuration given above would run the MSM for a 30 minute period from 1800 hours on day 112 of 1988 to 1830 on the same day. The sunspot number is zero.

To restart the MSM from 1830 (direct-access record number 3) and run until 2000 UT, the following MSMIN file would be used:

```
1988 112 66600
1988 112 72000
3
'MSM restart from 1830 on day 112'
0.0
```

Input file ENCHAN controls how many and which energy channels are calculated. The file set up is:

```
Number of energy channels to be modeled (maximum 30,integer)
Particle type 1 (integer), Energy 1 (floating point)
Particle type 2 (integer), Energy 2 (floating point)
. . .
Particle type n (integer), Energy n (floating point)
```

The acceptable particle type input is:

IMPLEMENTATION PROCEDURES - 4

1	electrons
2	H ⁺
3	O ⁺

The energy is at geosynchronous orbit in keV. The energies must be in ascending order, with particle types grouped together, which are checked within the program.

For example,

4	
1	40
1	55
2	10
2	15

This is to model 4 energy channels, the first two are electrons of 40 and 55 keV at geosynchronous orbit. The second two are H⁺ ions of 10 and 15 keV energy at geosynchronous orbit.

MAINTENANCE MANUAL

Maintenance Manual

2.1 System Application. The Magnetospheric Specification Model was developed as a set of algorithms to be incorporated into a real-time program for use in the U.S. Air Force Air Weather Service Space Forecast Center. It is intended to help Space Forecast Center personnel perform their mission of providing information, particularly during magnetospheric disturbances, to "customers" who operate spacecraft. The output of the model includes (but is not limited to):

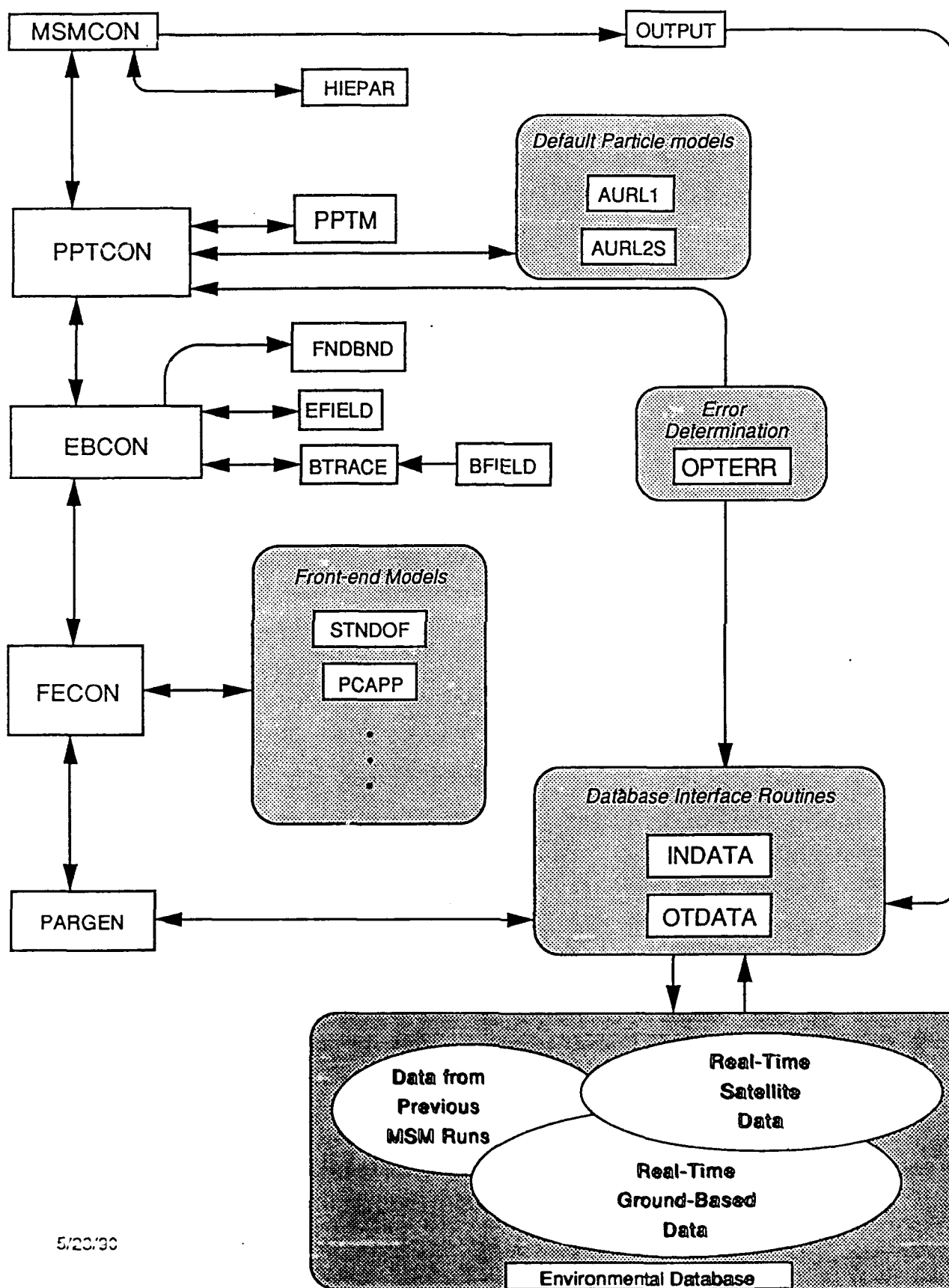
- 1) Fluxes of electrons and ions in the inner plasma sheet and the geosynchronous orbit region.
- 2) Energy fluxes and characteristic energies of electrons precipitated into the auroral ionosphere.

2.2 System Organization. The model uses separate ionospheric electric-field and magnetospheric magnetic field models to trace particle trajectories in the magnetosphere equatorial plane. The following flowchart shows the major components of the model and how they are related.

4.1 Conventions. None.

5.1.1 Description. See following pages.

MSM Control & Data Flow Diagram



MSMCON

a. Functions - The MSM control program MSMCON provides dimensioning of the arrays used throughout the program, and the initial, end and increments of time information for all other subroutines of the program. It also accesses the high energy particle tracing routines.

b. Input

ISTART(1)	Start Year (last two digits)
ISTART(2)	Start Day (julian day) January 1 is day 1.
ISTART(3)	Start Time (seconds)
IEND(1)	End Year (last two digits)
IEND(2)	End Day (julian day) January 1 is day 1.
IEND(2)	End Time (seconds)
IINC(1)	Program time step increment (year)
IINC(2)	Program time step increment (day)
IINC(3)	Program time step increment (seconds)
IRECEB	For restart, record number of previously calculated electric and magnetic field output data to use as input for current run
IRECPT	For restart, record number of previously calculated particle-trace output data to use as input for current run

c. Processing - following is a list of subroutines called and their major function:

PPTCON	Main control program for model
HIEPAR	High energy electron subroutine, currently a dummy routine pending computer code from sources outside Rice University.
OUTPUT	Output routine to interface with Space Forecast Center Environmental Database. Currently a dummy routine.

See the flowchart on the following page for the logic flow of the subroutine.

Global parameters for the run are set in this routine and are as follows:

LATDIM	Number of latitudinal grid lines (aliased as IDIM in some routines)
LTDIM	Number of local time (longitude) grid lines (aliased as JDIM in some routines)
IEDIM	Number of invariant energy species traced by the program
ITMDIM	Maximum number of major time steps per run of the program
NAUGEL	Number of elements in the augmented input array
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
KPLUS	Kp dimension of the augmented FLXMAT array ($KKPLUS = KPDIM + 2$)

ISPDIM	Number of plasma mass species (electrons, H^+ , O^+ , etc.)
IRDK	R dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sun spot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array

d. Output. None. All output is currently handled by other subroutines.

PPTCON

a. *Functions* - main control program for the electric and magnetic field models and the running of the particle trace algorithm.

b. Input -

ISTART	Start year, day, and time (sec)
IINC	Major time step increment (year, day, time)
IEND	Ending year, day, and time (sec)
LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
IEDIM	Number of particle species to be traced
ITMDIM	Maximum number of major time steps for this run
IRDK	R dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sun spot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array
IRDIM	R dimension of FLXMAT arrays
NRGDIM	Energy dimension of FLXMAT arrays
KPDIM	KP dimension of FLXMAT arrays
KPLUS	Kp dimension of the augmented FLXMAT array (KKPLUS = KPDIM + 2)
ISPDIM	Plasma mass species dimension

c. *Processing* - following is a list of subroutines called and their major function:

RDGRID	Subroutine to read grid system coordinates and calculate essential grid quantities
OUTP	Utility subroutine to print out array information
FLXNIT	Subroutine to calculate Kp-based particle number flux in units of particles/cm ² /s/sr/eV as function of R, energy and Kp
DTIME	Routine which returns current system time
EBCON	Executive routine to control calculation of electric and magnetic field models
EPREAD	Dummy routine to read satellite flux data - to be replaced by call to environmental data base when program is operational
SETALM	Subroutine to read the number of energy channels and their energies at geosynchronous orbit and calculate the energy invariant species needed to run the program
BNDSET	Subroutine to compute boundary plasma distribution
RDHDR	Subroutine to read the header record of the standard MSM disk file format
INITAL	Subroutine to compute initial plasma distribution
PPTM	Subroutine to trace particle trajectories from each grid point for each particle species back in time from time=TSTART to time=TSTOP

FLXDFL	Subroutine to calculate the energy-dependent ETA array from Kp and the location of the grid points by interpolating the empirical FLXMAT array - this routine is used as the MSM default model when full particle traces are not done
AURL1	Subroutine to determine default Hardy precipitating electron flux
AURL2S	Subroutine to determine default Hardy precipitating ion flux
PWRCAL	Subroutine to estimate precipitating electron energy flux poleward of the main MSM modeling region by comparing flux within modeling region with Hardy statistical values
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format
READ3D	Subroutine to read 3-d arrays from the standard MSM disk file format
RDVEC	Subroutine to read header vector from standard MSM disk file format
WRTVEC	Subroutine to write header vector from standard MSM disk file format
OPTERR	Subroutine to calculate standard deviation and error of the model flux and available observed geosynchronous satellite data

See the following flow-chart for the logical structure of the subroutine.

d. Output -

FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}(\text{eV})$)
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}(\#/\text{cm}^2\text{-s-ster-eV})$)
THRMAT	Array of empirical fluxes as function of R, energy, and KP for use as threshold fluxes. ($\log_{10}(\#/\text{cm}^2\text{-s-ster-eV})$)
CHID	Character header string (up to 80 characters)
ID	Integer header vector ID(1) = Year ID(2) = Day ID(3) = Hours ID(4) = Minutes ID(5) = Seconds ID(6) = Presently unused ID(7) = Presently unused ID(8) = First dimension of output array ID(9) = Second dimension of output array ID(10) = Third dimension of output array ID(11) = Time index L ID(12) - ID(20) = Presently unused
RID	Real header vector RID(1) = TIMTAG (seconds) RID(2) = KP RID(3) = Polar cap potential drop (kV)

	RID(4) = Time derivative of location of low-latitude edge of auroral precipitation (degrees/hour)
	RID(5) - RID(20) = Presently unused
SCTFRC	Assumed pitch angle scattering efficiency
IFLAV	Chemical species identifier
	1 means electrons
	2 means H ⁺ ions
	3 means O ⁺ ions
BNDMAX	Maximum I value of model boundary along each local time grid line during the duration of this run.
ALPHA	Latitudinal grid spacing vector
BETA	Longitudinal grid spacing vector
TETA	Colatitude of I grid lines (radians)
PHI	Non-rotated hour angle of J grid lines (radians)
BIR	Radial component of ionospheric magnetic field (nT)
SINI	SIN(Magnetic field inclination angle)
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
COLAT	Grid colatitude array (radians)
ALOCT	Grid local time array (radians eastward from noon)
V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts)
VNORTH	Northern hemisphere electric potential distribution (Volts)
VSOUTH	Southern hemisphere electric potential distribution (Volts)
ASOUTH	Southern hemisphere radius of ellipse measured in x (sunward) direction
BSOUTH	Southern hemisphere radius of ellipse measured in y (duskward) direction
DXS	Southern hemisphere radius of ellipse measured in y (duskward) direction
DYS	Southern hemisphere radius of ellipse measured in y (duskward) direction
ANORTH	Northern hemisphere radius of ellipse measured in x (sunward) direction
BNORTH	Northern hemisphere radius of ellipse measured in y (duskward) direction
DXN	Northern hemisphere radius of ellipse measured in y (duskward) direction
DYN	Northern hemisphere radius of ellipse measured in y (duskward) direction
VM	Flux tube volume ((Re/nT) ^{-2/3})
VMLOSS	Geometric factor used in calculating precipitation loss
BMIN	Equatorial magnetic field strength (nT)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
ALAM	Energy invariant vector (eV(Re/nT) ^{2/3})
ETABEG	Initial flux tube content distribution each time step (weber ⁻¹)
ETABND	Boundary flux tube content distribution (weber ⁻¹)
FLXBND	Flux at boundary (cm ² second) ⁻¹
BNDLOC	Location of outer boundary of detailed particle traces
ETA	Flux tube content array (weber ⁻¹)

FLUX	Flux (cm ² second) ⁻¹
EFLUX	Precipitation energy flux array (ergs/cm ² -sec)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec)
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
ALMDEL	Width of energy invariant channels
NAUGEL	Dimension of input arrays
AUGPAR	Augmented data array for input values
MODE	Logical variable telling which input parameters we have values for
RPP	Nominal radius of plasmapause defined as dipole mappine of equatorward edge of auroral zone to magnetospheric equatorial plane (Re)
THRSH	Vector giving full-traceback/default flux information
ERSHFT	Vector of error values
DKTIME	Array giving charge exchange decay times (seconds) for an array of radial distances, energies, sunspot numbers, and species

EBCON

a. Functions - EBCON calls routines to calculate the electric and magnetic field at our grid points for all time labels for this run.

b. Input -

ISTART	Start year, day, and time (sec)
IINC	Major time step increment (year, day, time)
IEND	Ending year, day, and time (sec)
COLAT	Grid colatitude array (radians)
ALOCT	Grid local time array (radians eastward from noon)
LATDIM	Number of latitudinal grid lines (aliased as IDIM in some routines)
LTDIM	Number of local time (longitude) grid lines (aliased as JDIM in some routines)
ITMDIM	Maximum number of major time steps per run of the program
IEDIM	Number of invariant energy species traced by the program
NAUGEL	Number of elements in the augmented input array
CHID	Character header string (up to 80 characters)
BIR	Radial component of ionospheric magnetic field (nT)
SINI	SIN(Magnetic field inclination angle)
IEMAX	Maximum number of energy channels
TETA	Colatitude of I grid lines (radians)

c. Processing - following are the subroutines called and their major function:

FECON	Subroutine to obtain time-normalized observational values
MLTSET	Subroutine to calculate time dependent ALOCT of rotating grid points
SETSCT	Subroutine to set species dependent pitch angle scattering efficiency
BTRACE	Subroutine to calculate magnetic field models appropriate for geophysical conditions as specified by standoff distance, equatorward edge of the auroral zone, Dst, tail collapse, and tilt angle
EFBNDY	Subroutine to calculate latitudes and widths of electric field model boundaries 2 and 3
EFIELD	Subroutine to return values of the potential on the northern hemisphere grid (VNORTH), on the southern hemisphere grid (VSOUTH), and the average of the two hemispheres (V)
OUTP	Utility subroutine to print out array information
FNDBND	Subroutine to compute time dependent outer boundary location
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
VMLSET	Subroutine to calculate VMLOSS array
WRT3D	Subroutine to write 3-d arrays into the standard MSM disk file format

d. Outputs -

ID	Integer header vector ID(1) = Year ID(2) = Day ID(3) = Hours ID(4) = Minutes ID(5) = Seconds ID(6) = Presently unused ID(7) = Presently unused ID(8) = First dimension of output array ID(9) = Second dimension of output array ID(10) = Third dimension of output array ID(11) = Time index L ID(12) - ID(20) = Presently unused
RID	Real header vector RID(1) = TIMTAG (seconds) RID(2) = KP RID(3) = Polar cap potential drop (kV) RID(4) = Time derivative of location of low-latitude edge of auroral precipitation (degrees/hour) RID(5) - RID(20) = Presently unused
VM	Flux tube volume (weber ⁻¹)
BMIN	Equatorial magnetic field strength (nT)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
V	Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts)
VNORTH	Northern hemisphere electric potential distribution (Volts)
VSOUTH	Southern hemisphere electric potential distribution (Volts)
ASOUTH	Southern hemisphere radius of ellipse measured in x (sunward) direction
BSOUTH	Southern hemisphere radius of ellipse measured in y (duskward) direction
DXS	Southern hemisphere radius of ellipse measured in y (duskward) direction
DYS	Southern hemisphere radius of ellipse measured in y (duskward) direction
ANORTH	Northern hemisphere radius of ellipse measured in x (sunward) direction
BNORTH	Northern hemisphere radius of ellipse measured in y (duskward) direction
DXN	Northern hemisphere radius of ellipse measured in y (duskward) direction
DYN	Northern hemisphere radius of ellipse measured in y (duskward) direction
BNDLOC	Location of outer boundary of detailed particle traces
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ITMMAX	Maximum number of time labels

R	Array giving radial distance of grid pts in magnetospheric equatorial plane (R_e)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VMLOSS	Geometric factor used in calculating precipitation loss
SCTFRC	Assumed pitch angle scattering efficiency
AUGPAR	Augmented data array for input values
MODE	Logical variable telling which input parameters we have values for
RPP	Nominal radius of plasmapause defined as dipole mapping of equatorward edge of auroral zone to magnetospheric equatorial plane (R_e)

BLOCK DATA

a. Functions - Block data containing the index of data elements and connections between the logical unit number and names. Sets I, J, and energy indices for locations and species to be traced.

b. Input - none.

c. Processing - No other subroutines are called.

The block data contains the data element assignments (for example for the AUGPAR vector) and connects the logical unit number to the name within the program.

d. Output - the following assignments are made:

Elements in the input data array

1	Year
2	Day
3	Seconds of day
4	Minute
5	Seconds
6	Kp
7	Dst (nT)
8	Equatorward edge of auroral zone at midnight (degrees)
9	Low latitude boundary of auroral zone (degrees)
10	Equatorward edge of Region 2 currents (degrees)
11	Currently unused
12	Bx component of interplanetary magnetic field (nT)
13	By component of interplanetary magnetic field (nT)
14	Bz component of interplanetary magnetic field (nT)
15	Magnetotail collapse parameter
16	X component of geosynchronous magnetic field data (nT)
17	Y component of geosynchronous magnetic field data (nT)
18	Z component of geosynchronous magnetic field data (nT)
19	Currently unused
20	Currently unused
21	Solar wind velocity (km/sec)
22	Solar wind density (prot/cm ³)
23	Tilt angle (degrees)
24	Standoff distance (Re)
25	Cross polar cap potential drop (kV)
26	Electric field pattern type
27	Time rate of change of Dst (nT/hour)

28 Time rate of change of the equatorward edge of the auroral oval at midnight (degrees/hour)

Elements in the logical unit assignment:

9	LUERR - Error output
13	LUPPT - Not currently used
14	LUCORD - Coordinate data
15	LUPRNT - Output to printer
16	LUIDAT - Not currently used
17	LUHDYE - Coefficients to calculate electron precipitating energy and number flux in default model
20	LUDK - charge exchange decay parameter file
21	LUETAB - ETABND file (boundary flux tube content distribution)
22	LUEBEG - ETABEG file (initial flux tube content distribution each time step)
23	LUSHFT - error correction factor file
24	LUPION - Not currently used
25	LUEAVG - EAVG file (average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺)
26	LUFLSM - FLXSUM file (total precipitation energy flux array (integrated over species for (1) electrons, (2) H ⁺ and (3) O ⁺))
27	LUFLX - FLUX file (particle flux array)
28	LUEFLX - EFLUX file (precipitation energy flux array)
35	LUEFLD - V file (electric potential distribution on grid (average of VNORTH and VSOUTH))
36	LUBFLD - VM file (flux tube volume)
37	LUBMIN - BMIN file (equatorial magnetic field strength (nT))
38	LUXMIN - XMIN file (GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane)
39	LUYMIN - YMIN file (GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane)
40	LUZMIN - ZMIN file (GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane)
41	LUALOC - ALOCT file (grid local time array (radians eastward from noon))
42	LUCOLT - COLAT file (grid colatitude array (radians))
43	LUBNDL - BNDLOC file (location of outer boundary of detailed particle traces)
44	LUPDAT - AUGPAR vector (augmented data array for input values)
45	LUVN - VNORTH file (northern hemisphere electric potential distribution)
46	LUVS - VSOUTH file (southern hemisphere electric potential distribution)
47	LIONEG - Coefficients to calculate ion precipitating energy flux in default model
48	LIONNO - Coefficients to calculate ion number flux in default model
49	LUENCH - File used to input number and range of energy channels to model

Other assignments made:

IBEG	Starting I value, currently set to 1
ISTOP	Ending I value, currently set to 53
JBEG	Starting J value, currently set to 3
JSTOP	Ending J value, currently set to 50
IEBEG	Starting energy value index, currently set to 1

EMODEL

a. Functions - This subroutine specifies the electric potential at each grid point in the modelling region for the northern hemisphere, southern hemisphere and the average of both hemispheres.

b. Inputs

IPATT	Polar cap pattern type (from Heppner-Maynard empirical model)
PCP	Cross polar cap potential (Volts)
DEQDT	Time rate of change of the latitude of the equatorward edge of the auroral zone at midnight (degrees/hour)
A	Radius of ellipse measured in X(sunward) direction
B	Radius of ellipse measured in Y(duskward) direction
DX	Sunward displacement of coordinate system center from pole
DY	Duskward displacement of coordinate system center from pole
COLAT	Grid colatitude array (radians)
ALOCT	Grid local time array (radians eastward from noon)
MODE	Not currently used
LATDIM	Number of latitudinal grid lines
LTDIM	Number of local time (longitude) grid lines
JWRAP	Number of points of overlap in J (longitude) direction
ICNTRL	= 1 means different formulas are used in regions 1, 2, 3, = 2 means Heppner-Maynard formula is used for all latitudes , but scaled to externally specified cross polar cap potential drop and ellipse parameters = -1 means Heppner-Maynard formula is used for all latitudes, unscaled. = -2 should not be used. It is the form used by the Rice Convection Model
ITOP	= 1 means the ellipse parameters A, B, DX, and DY for the polar cap boundary are computed internally in the subroutine. This is the normal mode for the MSM.

c. Processing - following is a list of the subroutines called and their major function:

INPUT	Get input needed for computation of Heppner-Maynard model
THET	Function that gives equation for the electric field boundary ellipse in flat polar coordinates
EPOT	Using scaled Heppner-Maynard model returns potential at a specific point
LOW	Returns potential at a specific location in low latitude region
AURORA	Returns potential at a specific location in auroral region
REG1	Returns potential at a specific location in Region 1

A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

V Electric potential disbritution on grid (Volts)

EPOT

a. Functions - This subroutine uses the scaled Heppner-Maynard model to return the electric potential at a specific point.

b. Input -

COEF	Coefficients from digitization of Heppner-Maynard electric field model
TLAT	Colatitude of point on boundary to be calculated (radians)
TLON	Magnetic local time of point on boundary to be calculated (radians)
NTAPE	Unit number from which to read Heppner-Maynard coefficients
IPATT	Heppner-Maynard pattern type
NNMAX	Number of coefficients to use in calculation
A	Radius of ellipse measured in x (sunward) direction
B	Radius of ellipse measured in y (duskward) direction
DX	Radius of ellipse measured in y (duskward) direction
DY	Radius of ellipse measured in y (duskward) direction
XCO	Electric field model coefficients
AHM	Radius of ellipse measured in x (sunward) direction from Heppner-Maynard Model
BHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
DXHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
DYHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
VMIN	Computed electric potential minimum from Heppner-Maynard Model
VMAX	Computed electric potential maximum from Heppner-Maynard Model
PCP	Cross-polar-cap potential drop
ICNTRL	= 1 means different formulas are used in regions 1, 2, 3, = 2 means Heppner-Maynard formula is used for all latitudes , but scaled to externally specified cross polar cap potential drop and ellipse parameters = -1 means Heppner-Maynard formula is used for all latitudes, unscaled. = -2 should not be used. It is the form used by the Rice Convection Model

c. Processing - A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

VALUE	Electric potential at point on boundary a (Volts)
-------	---

LOW

a. Functions - This subroutine returns the electric potential at a specification location in the low latitude region of the electric field model.

b. Input -

COL	Colatitude of point on boundary to be calculated (radians)
ALO	Magnetic local time of point on boundary to be calculated (radians)
DEQDT	Time rate of change of equatorward edge of auroral zone (degrees/hour)
AEQEDG	Radius of ellipse 3 measured in x (sunward) direction
DXEQEG	Offset of ellipse 3 in x direction
VBAR	Average potential at low latitudes
G	Weighting factor

c. Processing - A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

VLOW	Electric potential of a point in low latitude region (Volts)
------	--

AURORA

a. Functions - This subroutine returns the electric potential at a specific location in the auroral zone region of the electric field model.

b. Input -

COL	Colatitude of point on boundary to be calculated (radians)
ALO	Magnetic local time of point on boundary to be calculated (radians)
VBAZ	Auroral-zone contribution to the potential at boundary b
TB	Location of boundary b for specific local time
TC	Location of boundary c for specific local time
J	Local time index
JMAX	Maximum number of points in J direction
JWRAP	Number of points of overlap in J direction

c. Processing - A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

VAZ	Electric potential for a point in the auroral zone (Volts)
-----	--

REG1

a. Functions - This subroutine returns the electric potential for a specific point in Region 1 of the electric field model.

b. Input -

COL	Colatitude of point on boundary to be calculated (radians)
AL	Magnetic local time of point on boundary to be calculated (radians)
TA	Location of boundary a for specific local time
TB	Location of boundary b for specific local time
VAA	Electric potential for point on boundary a
VBB	Electric potential for point on boundary b
DVAA	Derivative of electric potential for point on boundary a
DVBB	Derivative of electric potential for point on boundary b

c. Processing - A concise description of the electric field model and its components is given in Section 2.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

V1	Electric potential for a point in Region 1 (Volts)
----	--

THET

a. Function - This function subprogram gives the equation for an electric field boundary ellipse in flat polar coordinates.

b. Input -

AA	Radius of ellipse measured in X(sunward) direction
BB	Radius of ellipse measured in Y(duskward) direction
XC	Sunward displacement of coordinate system center from pole
YC	Duskward displacement of coordinate system center from pole
PHI	Magnetic local time

c. Processing - This function gives the solution to the equation for an ellipse in flat polar coordinates. It specifies colatitude as a function of local time.

d. Output -

THET	Colatitude of input ellipse parameters and local time
------	---

INPUT

a. Function - This subroutine reads in the necessary input for the computation of the Heppner-Maynard electric field model.

b. Input -

NTAPE Unit number from which to read Heppner-Maynard coefficients

c. Processing - This subroutine reads from logical unit NTAPE the coefficients necessary for the Heppner-Maynard electric field model.

d. Output -

COEF	Coefficients from digitization of Heppner-Maynard electric field model
IPATT	Polar cap pattern type (from Heppner-Maynard empirical model)
XCO	E field coefficients
AHM	Radius of ellipse measured in x (sunward) direction from Heppner-Maynard Model
BHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
DXHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
DYHM	Radius of ellipse measured in y (duskward) direction from Heppner-Maynard Model
VMIN	Computed electric potential minimum from Heppner-Maynard Model
VMAX	Computed electric potential maximum from Heppner-Maynard Model

BTRACE

a. Functions - Subroutine to calculate magnetic field models for appropriate geophysical conditions as specified by the standoff distance, equatorward edge of the auroral zone, Dst, tail collapse parameter, and tilt angle. Calculates the output on the MSM spatial grid by interpolating between pre-computed magnetic field models.

b. Input -

LATDIM	Number of latitudinal grid lines
LTDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program
L	Current temporal grid index
FSTOFF	Standoff distance (Re)
FEQEDG	Equatorward edge of aurora (degrees)
FDST	Dst (nT)
FCLPSE	Collapse parameter
FTILT	Tilt angle (degrees)
ALOCT	Local time hour angle (measured east from noon)
TETA	Colatitude of I grid lines (radians)

c. Processing - following are the subroutines called and their major functions:

GETMAT	Subroutine to find the magnetic field matrices needed for interpolation in order to represent current conditions
RMVBSH	Subroutine to extrapolate magnetic field matrices into extremely stretched or open regions
BFGYRO	Subroutine to rotate fixed grid magnetic field arrays to match rotating coordinate system

This subroutine determines which of the pre-computed magnetic field matrices are needed to match the geophysical conditions specified. After the matrices are retrieved and stored into a work array (WORK(LATDIM,LTDIM,2,2,2,2)) they are interpolated using a multilinear interpolation scheme given by Press et al. in Numerical Recipes, Cambridge University Press, 1987, pp 95-97. The arrays are then rotated so that they have the same local time orientation as the main program.

d. Output -

VM	Flux tube volume $((Re/nT)^{-2/3})$
BMIN	Equatorial magnetic field strength (nT)
XMIN	GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
YMIN	GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)

ZMIN	GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane

BFGYRO

a. Functions - Subroutine to rotate fixed grid magnetic field arrays to match the rotating grid system of the main program.

b. Inputs -

LATDIM	Number of latitudinal grid lines
LTDIM	Number of local time (longitude) grid lines
ARRAY	Unrotated magnetic field values called sequentially for the following quantities: VM - Flux tube volume $((Re/nT)^{-2/3})$ BMIN - Equatorial magnetic field strength (nT) XMIN - GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re) YMIN - GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re) ZMIN - GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
WORK	Internal working array
ALOCT	Local time hour angle (measured east from noon)

c. Processing - No other subroutines are called.

This routine uses a linear interpolation scheme to determine how much the magnetic field matrices should be rotated (F). It applies this factor to the input matrices so that the magnetic local time of the magnetic field matrices and the main program agree.

d. Output -

ARRAY	Rotated magnetic field values for the following quantities: VM - Flux tube volume $((Re/nT)^{-2/3})$ BMIN - Equatorial magnetic field strength (nT) XMIN - GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re) YMIN - GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re) ZMIN - GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)
-------	--

PPTM

a. Functions - This subroutine traces particle trajectories from each grid point for each particle species back in time from time=TSTART to time=TSTOP.

b. Inputs

TSTART	Time at which to begin particle traceback (seconds)
TSTOP	Time at which to end particle traceback (seconds)
DTNOM	Nominal time step to begin traces. DTNOM < 0 to trace back in time. PPTM will adjust time step as needed
SCTFRC	Assumed pitch angle scattering efficiency
IFLAV	Chemical species identifier 1 means electrons 2 means H ⁺ ions 3 means O ⁺ ions
BNDMAX	Maximum I value of model boundary along each local time grid line during the duration of this run.
ALPHA	Latitudinal grid spacing vector
BETA	Longitudinal grid spacing vector
BIR	Radial component of ionospheric magnetic field (nT)
ITMMAX	Number of temporal grid points used
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
IEDIM	Number of invariant energy species traced by the program
ITMDIM	Maximum number of major time steps per run of the program
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
KPPLUS	KP dimension of augmented FLXMAT array (KPPLUS=KPDIM+2)
ISPDIM	Number of plasma mass species (electrons, H ⁺ , O ⁺ , etc.)
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated (log ₁₀ (eV))
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values (log ₁₀ (#/cm ² -s-ster-eV))
THRMAT	Array of empirical fluxes as function of R, energy, and KP for use as threshold fluxes. (log ₁₀ (#/cm ² -s-ster-eV))
TIMTAG	Vector giving times at which E and B parameters are calculated
ALAM	Energy invariant vector (eV(Re/nT) ^{2/3})
ALMDEL	Width of energy invariant channels
VM	Flux tube volume ((Re/nT) ^{-2/3})
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane

VMLOSS	Geometric factor used in calculating precipitation loss
V	Electric potential distribution on grid (Volts)
ETABEG	Initial flux tube content distribution each time step (weber ⁻¹)
ETABND	Boundary flux tube content distribution (weber ⁻¹)
BNDLOC	Location of outer boundary of detailed particle traces
LL	Current temporal grid index
RPP	Nominal radius of plasmapause defined as dipole mapping of equatorward edge of auroral zone to magnetospheric equatorial plane (Re)
FKP	Current Kp value
IEMAX	Number of energy channels in use
THRSH	Vector giving full-traceback/default flux information
TSHENG	Threshold value to switch from full-traceback to default models - current set at > 100 keV electrons
TSHING	Threshold value to switch to default models for ions - currently set at 50 keV
IRDK	R dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sun spot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array
DKTIME	Array giving charge exchange decay times (seconds) for an array of radial distances, energies, sunspot numbers, and species

c. Processing - following are the subroutines called and their major function

SETREF	Subroutine to set upper limit reference flux
OUTP	Utility routine to print our array information
TNORML	Utility function subprogram to calculate run-normalized time
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
WKRATE	Function subprogram to evaluate weak precipitation loss rate
MOVER	Subroutine to advance one particle location one time step using 4th-order Runge-Kutta algorithm with 5th order correction
MODFIX	Subroutine to adjust BBJ to be between 2 and JDIM-1
TCHK	Subroutine to check whether time boundary was crossed
BNDCHK	Subroutine to determine whether particle has crossed boundary of calculation
FLXVAL	Function subprogram to calculate Kp-dependent flux value a R=RR for particles of energy ENRG and Kp=FKP by interpolating FLXMAT array

See the following flow chart for the logic of the subroutine.

d. Output

ETA	Flux tube content array (weber ⁻¹)
-----	--

EFLUX	Precipitation energy flux array (ergs/cm ² -sec)
FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec)
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV)

BNDCHK

a. Functions - This subroutine determines whether a particle has crossed the spatial boundary of the model calculation.

b. Input -

T	Time
IE	Energy channel
BBI	I location at end of time step
BBJ	J location at end of time step
BIOLD	I location at beginning of time step
BJOLD	J location at beginning of time step
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ETABND	Boundary flux tube content distribution (weber ⁻¹)
BNDMAX	Maximum I value of model boundary along each local time grid line during the duration of this run.
BNDLOC	Location of outer boundary of detailed particle traces
JDIM	Number of local time (longitude) grid lines
IEDIM	Number of invariant energy species traced by the program
ITMDIM	Maximum number of major time steps per run of the program
ITMMAX	Number of temporal grid points used
JWRAP	Number of points of overlap in J direction

c. Processing - following are the subroutines called and their major function:

 TNORML Utility function subprogram to calculate run-normalized time

Using BNDMAX, the I value of the boundary, the time-normalized I (BBI) and J (BBJ) input values are compared to see if the particle has passed out of the outer boundary of the calculation region. If the particle has passed the boundary, the ETA value at that point on the boundary is calculated and returned.

d. Output -

ETANIL	If boundary is crossed, ETANIL is the value of ETA at the boundary crossing point (weber ⁻¹)
IEXIT	If the boundary is crossed, IEXIT = 2

DVEFDI

a. Function - Function subprogram to compute the I derivative of the effective potential at grid point (I,J) at normalized time BT.

b. Input -

I	I coordinate of the grid point
J	J coordinate of the grid point
BT	Normalized time
AALAM	Energy invariant vector ($eV(Re/nT)^{2/3}$)
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume ($(Re/nT)^{-2/3}$)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - No other subroutines are called.

Standard central-difference formula is used to differentiate the effective potential (potential energy - kinetic energy) in the I direction.

d. Output -

DVEFDI Derivative of effective potential at point (I,J,BT)

DVEFDJ

a. Function - Function subprogram to compute the J derivative of the effective potential at grid point (I,J) at normalized time BT.

b. Input -

I	I coordinate of the grid point
J	J coordinate of the grid point
BT	Normalized time
AALAM	Energy invariant vector ($eV(Re/nT)^{2/3}$)
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume ($(Re/nT)^{-2/3}$)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - No other subroutines are called.

Standard central-difference formula is used to differentiate the effective potential (potential energy - kinetic energy) in the J direction.

d. Output -

DVEFDJ Derivative of effective potential at point (I,J,BT)

G3NTRP

a. Functions - Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

b. Input -

A	Array to be interpolated
IMAX	I dimension of array A
JMAX	J dimension of array A
KMAX	K dimension of array A
BI	Floating point value to interpolate in I dimension
BJ	Floating point value to interpolate in J dimension
BK	Floating point value to interpolate in K dimension

c. Processing - No other subroutines are called.

This subroutine uses the standard linear interpolation technique to interpolate an array in 3 dimensions.

d. Output -

G3NTRP	Interpolated point from array A
--------	---------------------------------

MODFIX

a. Functions - Subroutine to adjust J value to be within array bounds.

b. Input -

BBJ	J value to be adjusted
JDIM	J dimension of array
JWRAP	Number of points of overlap in J direction

c. Processing - No other subroutines are called.

The subroutine tests to see if the value is within the array bounds and adjusts the value if it is not.

d. Output -

BBJ	Adjusted J value
-----	------------------

MOVER

a. Functions - Subroutine to advance one particle location one time step using 4th-order Runge-Kutta algorithm with 5th order correction.

b. Input -

T	Time parameter
DTTRY	Time step to try < 0 = moves particles back in time > 0 = moves particles forward in time
BIOLD	Current I location of particle
BJOLD	Current J location of particle
EPS	Accuracy parameter
BBISCL	I error scaling parameter
BBJSCL	J error scaling parameter
AALAM	Energy invariant vector ($eV(Re/nT)^{2/3}$)
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume ($(Re/nT)^{2/3}$)
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ALPHA	Latitudinal grid spacing vector
BETA	Longitudinal grid spacing vector
BIR	Radial component of ionospheric magnetic field (nT)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - following are the subroutines called and their major functions:

RK4 4th order Runge-Kutta routine

This routine is based on the discussion in Numerical Recipes, Press et al, on pages 550-560 in the chapter on integration of ordinary differential equations.

d. Output -

BBI	Updated I location of particle
BBJ	Updated J location of particle
DTUSED	Time step actually used
DTNEXT	Estimated next stepsize

RK4

a. Functions - This subroutine integrates an ordinary differential equation using a 4th order Runge-Kutta scheme.

b. Inputs -

T	Time parameter
DT	Time step < 0 = moves particles back in time > 0 = moves particles forward in time
BBIIN	Current I location of particle
BBJIN	Current J location of particle
AALAM	Energy invariant vector ($eV(Re/nT)^{2/3}$)
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume ($(Re/nT)^{-2/3}$)
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ALPHA	Latitudinal grid spacing vector
BETA	Longitudinal grid spacing vector
BIR	Radial component of ionospheric magnetic field (nT)
IDIM	Number of latitudinal grid lines
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - following are the subroutines called and their major function:

VLOCTY	Subroutine to compute I and J components of particle velocity with energy invariant AALAM at location (BI,BJ) at normalized time BT
--------	---

This algorithm comes from Abramowitz, Milton and Stegun, Irene A., 1964, Handbook of Mathematical Function, National Bureau of Standards, 1970, p. 896, section 25.5.10. It is also referenced in Numerical Recipes by Press et al.

d. Output

BBIOUT	Computed I location at end of time step
BBJOUT	Computed J location at end of time step

TCHK

a. Functions - This subroutine checks whether a particle has crossed a time boundary for this time step.

b. Input -

T	Time
TSTOP	Time at which to end particle traceback
DTNOM	Nominal time step to begin traces
BBI	I location of particle
BBJ	J location of particle
IEUSE	Energy index of particle
ETABEG	Initial flux tube content distribution each time step (weber ⁻¹)
IDIM	Number of latitudinal grid lines
JDIM	Number of local time (longitude) grid lines
IEDIM	Number of invariant energy species traced by the program

c. Processing - following are the subroutines called and major functions:

G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
--------	--

If time boundary is crossed, the ETA at the location (BBI,BBJ) is interpolated.

d. Output -

ETANIL	ETABEG value at location (BBI,BBJ) at time T for energy species IEUSE if particle has crossed time boundary (weber ⁻¹)
IEXIT	= 1 if particle has crossed time boundary

TNORML

a. Functions - Utility function subprogram to calculate run-normalized time.

b. Input -

TT	Time (seconds)
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
ITMMAX	Number of temporal grid points used

c. Processing - No other subroutines are called.

Using the run times given in TIMTAG, the time is located. Using standard linear interpolation, the correction factor is calculated.

d. Output -

TNORML	Non-integer time index corresponding to time TT
--------	---

VLOCTY

a. Functions - This subroutine computes the I and J components of particle velocity for a particle with energy invariant AALAM at location (BBI,BBJ) at normalized time BT.

b. Input -

BBI	I location of particle
BBJ	J location of particle
BT	Normalized time
AALAM	Energy invariant vector ($eV(Re/nT)^{2/3}$)
ALPHA	Latitudinal grid spacing vector
BETA	Longitudinal grid spacing vector
BIR	Radial component of ionospheric magnetic field (nT)
V	Electric potential distribution on grid (Volts)
VM	Flux tube volume $((Re/nT)^{2/3})$
IDIM	Number of latitudinal grid lines
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - following are the subroutines called and major functions:

G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
--------	--

This subroutine computes the I and J components of the particle's ExB drift.

d. Output -

DBIDT	I velocity of particle (I units/sec)
DBJDT	J velocity of particle (J units/sec)

RDGRID

a. Functions - Subroutine to read grid system coordinates and calculate essential grid quantities.

b. Input - Read from input file 'COORD'

ID	Eight character identifier
OFFSET	Offset of center of coordinate system from magnetic pole, currently zero
DLAM	Latitudinal grid spacing constant
DPSI	Local time grid spacing constant
RI	Radius of earth + ionosphere (km)
RE	Radius of earth (km)
ALPHA	Latitudinal grid spacing vector
BETA	Longitudinal grid spacing vector
TETA	Colatitude of I grid lines (radians)
PHI	Non-rotated hour angle of J grid lines (radians)

c. Processing - No other subroutine are called.

d. Additional outputs -

BIR	Radial component of ionospheric magnetic field (nT)
SINI	SIN(Magnetic field inclination angle)
COLAT	Grid colatitude array (radians)

MLTSET

a. Functions - This subroutine calculates the time dependent magnetic local time of rotating grid points.

b. Input -

ISEC	Time from start of run (seconds)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines

c. Processing - No other subroutines are called.

The subroutine calculates the change in the hour angle of the J grid lines (DPHI) and the time rate of change of the hour angle of the J grid lines (DPHIDT) to compute the time dependent magnetic local time grid

d. Output -

ALOCT	Time dependent magnetic local time grid
-------	---

VMLSET

a. Functions - This subroutine sets up the time-dependent VMLOSS array

b. Input -

ITMCUR	Current time label
VM	Flux tube volume $((R_e/nT)^{-2/3})$
BIR	Radial component of ionospheric magnetic field (nT)
SINI	SIN(Magnetic field inclination angle)
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - No other subroutines are called.

d. Output -

VMLOSS	Geometric factor used in calculating precipitation loss
--------	---

FNDBND

a. Functions - This subroutine computes the time dependent outer model boundary location.

b. Input -

ITMCUR	Current time label
COLAT	Grid colatitude array (radians)
ALOCT	Time dependent magnetic local time grid (radians eastward from noon)
A2	Radius of ellipse 2 measured in x (sunward) direction
B2	Radius of ellipse 2 measured in y (duskward) direction
DX2	Offset of ellipse 2 measured in y (duskward) direction
DY2	Offset of ellipse 2 measured in y (duskward) direction
IDIM	Number of latitudinal grid lines)
JDIM	Number of local time (longitude) grid lines
ITMDIM	Maximum number of major time steps per run of the program

c. Processing - following are the subroutines called and their major function:

THET	Function that gives the equation for an electric field boundary ellipse in flat polar coordinates
------	---

For each J (longitudinal) value, the location of the outer boundary is calculated. Then the colatitude array (COLAT) is searched to find the pair of I values that encompass the boundary. The location is then linearly interpolated and placed in the array BNDLOC.

d. Output -

BNDLOC	Location of outer boundary of detailed particle traces
--------	--

OUTP

a. Functions - This is utility subroutine to print out array information.

b. Input -

R	Array to be output
ISIZE	I dimension of R array
JSIZE	J dimension of R array
IBEG	Initial I value to be output
IEND	Final I value to be output
IINC	I value increment for output
JBEG	Initial J value to be output
JEND	Final J value to be output
JINC	J value increment for output
XSCALE	= 0.0 scale is calculated by subroutine to give best display. All elements or array are divided by scale before being output ≠ 0.0 scale is set by user
TITLE	Character string that identifies the array being output
NTP	Output unit number
NCOL	Number of columns of output device (usually either 80 or 132)

c. Processing - No other subroutines are called.

This subroutine takes any 2-dimensional array and formats the output.

d. Output - The array is output to the device NTP.

AURL1 (also FNDHDY, ADDHDY, FSUM)

a. Function - These subroutines determine the default Hardy precipitating electron flux.

b. Input -

FKP	Current Kp value
COLAT	Grid colatitude array (radians)
ALOC	Time dependent magnetic local time grid (radians)
LATDIM	Number of latitudinal grid lines)
LTDIM	Number of local time (longitude) grid lines

c. Processing - following are the subprograms called and their major function:

FNDHDY	Subroutine to find and read the pertinent Fourier coefficients for the Hardy electron model
ADDHDY	Subroutine to calculate the Hardy model electron flux value for a given point
FSUM	Function subprogram to determine sum of a Fourier series

These subroutines were supplied by Dr. David Hardy at the Air Force Geophysics Laboratory, Bedford, Mass.

d. Output -

AEFLUX	The integral electron energy flux (ergs/cm ² -sec)
AEMEAN	The average energy (eV)

WRT3D

a. Functions - This subroutine writes 3-d arrays to disk in the standard MSM disk file format.

b. Input -

LUN	Logical unit number on which to write file
IRECMX	Logical record number at which to write data
ID	Integer header vector
	ID(1) = Year
	ID(2) = Day
	ID(3) = Hours
	ID(4) = Minutes
	ID(5) = Seconds
	ID(6) = Presently unused
	ID(7) = Presently unused
	ID(8) = First dimension of output array
	ID(9) = Second dimension of output array
	ID(10) = Third dimension of output array
	ID(11) = Time index L
	ID(12) - ID(20) = Presently unused
RID	Real header vector
	RID(1) = TIMTAG (seconds)
	RID(2) = KP
	RID(3) = Polar cap potential drop (kV)
	RID(4) = Time derivative of location of low-latitude edge of auroral precipitation. (degrees/hour)
	RID(5) - RID(20) = Presently unused
CHID	Character header string (up to 80 characters)
ARRAY	3-d array to be written
IDIM	I dimension of ARRAY
JDIM	J dimension of ARRAY
KDIM	K dimension of ARRAY
ITMDIM	Maximum number of major time steps for this run

c. Processing - No other subroutines are called.

This routine takes the desired array, opens the appropriate file, writes the array to the file starting with the specified logical record number, and closes the file.

d. Output - The array is written to unit number LUN.

RDHDR

a. Functions - This subroutine reads the header record of the standard MSM disk file

b. Input -

LUN	Logical unit number on which to write file
FILNAM	Character string containing file name
IRECMX	Record number to begin read function
IDIM	I dimension
JDIM	J dimension
ITMDIM	Maximum number of major time steps for this run

c. Processing - No other subroutines are called.

This routine opens the appropriate file, reads the header information of the file starting with the specified record number, and closes the file.

d. Output -

ID	Integer header vector ID(1) = Year ID(2) = Day ID(3) = Hours ID(4) = Minutes ID(5) = Seconds ID(6) = Presently unused ID(7) = Presently unused ID(8) = First dimension of output array ID(9) = Second dimension of output array ID(10) = Third dimension of output array ID(11) = Time index L ID(12) - ID(20) = Presently unused
RID	Real header vector RID(1) = TIMTAG (seconds) RID(2) = KP RID(3) = Polar cap potential drop (kV) RID(4) = Time derivative of location of low-latitude edge of auroral precipitation (degrees/hour) RID(5) - RID(20) = Presently unused
CHID	Character header string (up to 80 characters)

READ3D

a. Function - This subroutine reads a record from the standard MSM disk file.

b. Input -

LUN	Logical unit number from which to read file
FILNAM	Character string containing file name
LREC	Logical record number to read direct access data set
IDIM	1st dimension of ARRAY
JDIM	2nd dimension of ARRAY
KDIM	3rd dimension of ARRAY
ITMDIM	Maximum number of major time steps for this run

c. Processing - No other subroutines are called.

This routine opens the appropriate file, reads the array from the file starting with the specified record number, and closes the file.

d. Output -

ID	Integer header vector ID(1) = Year ID(2) = Day ID(3) = Hours ID(4) = Minutes ID(5) = Seconds ID(6) = Presently unused ID(7) = Presently unused ID(8) = First dimension of output array ID(9) = Second dimension of output array ID(10) = Third dimension of output array ID(11) = Time index L ID(12) - ID(20) = Presently unused
RID	Real header vector RID(1) = TIMTAG (seconds) RID(2) = KP RID(3) = Polar cap potential drop (kV) RID(4) = Time derivative of location of low-latitude edge of auroral precipitation (degrees/hour) RID(5) - RID(20) = Presently unused
CHID	Character header string (up to 80 characters)
ARRAY	3-d array to be read

SETALM

a. Functions - This subroutine reads the number of energy channels and their energies at geosynchronous orbit and calculates the energy invariant species corresponding to these energies.

b. Input - Read from file 'ENCHAN'

IFLAV	Chemical species identifier
	1 means electrons
	2 means H ⁺ ions
	3 means O ⁺ ions
ENCHNL	Energy channel at geosynchronous orbit (eV)

c. Processing - No other subroutines are called.

The input data must be in eV with energy increasing within each chemical species. The parameter VM6 is the typical value for (flux tube volume)^{-2/3} at geosynchronous orbit.

d. Output -

ALAM	Energy invariant vector (eV(Re/nT) ^{2/3})
IEMAX	Maximum number of energy channels
IEDIM	Number of invariant energy species traced by the program
IFLAV	Chemical species identifier
	1 means electrons
	2 means H ⁺ ions
	3 means O ⁺ ions
ALMDEL	Width of energy invariant channels

BNDSET

a. Functions - This subroutine computes the boundary plasma distribution.

b. Input -

LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
IEDIM	Number of particle species to be traced
ITMDIM	Maximum number of major time steps for this run
ITMMAX	Actual number of time steps for this run
IRDIM	R dimension of FLXMAT arrays
NRGDIM	Energy dimension of FLXMAT arrays
KPDIM	KP dimension of FLXMAT arrays
KPPLUS	Augmented KP dimension (=KPDIM+2)
ISPDIM	Plasma mass species dimension
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}(\text{eV})$)
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}(\#/\text{cm}^2\text{-s-ster-eV})$)
NAUGEL	Number of elements in the augmented input array
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
BNDLOC	Location of outer boundary of detailed particle traces
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
AUGPAR	Augmented data array for input values
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VM	$(\text{Flux tube volume})^{-2/3} ((\text{Re}/nT)^{-2/3})$
ETA	Flux tube content array (weber^{-1})
ALAM	Energy invariant vector ($\text{eV}(\text{Re}/nT)^{2/3}$)
ALMDEL	Width of energy invariant channels
IEMAX	Maximum number of energy channels

c. Processing - No other subroutines are called.

The background for this subroutine is contained in Section 2.5.5 of Contract #F19628-87-0001 Final Report.

d. Output

ETABND Boundary flux tube content distribution (weber^{-1})
FLXBND Flux values at boundary ($\text{cm}^2 \text{ sec}^{-1}$)

SETSCT

a. Functions - Subroutine to set species-dependent pitch angle scattering efficiency.

b. Input -

L	Time label
ISEC	Time (seconds)
IEDIM	Number of particle species to be traced
LTDIM	Number of local time (longitudinal) grid spaces
ITMDIM	Maximum number of major time steps for this run
LATDIM	Number of latitudinal grid spaces
FKP	Kp value for current time
ALAM	Energy invariant vector ($eV(Re/nT)^{2/3}$)
ALOCT	Grid local time array (radians eastward from noon)
IEMAX	Maximum number of energy channels

c. Processing - No other subroutines are called.

This subroutine sets the species-dependent pitch angle scattering efficiency. The rationale for this subroutine is discussed in Section 2.6.3 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

SCTFRC	Assumed pitch angle scattering efficiency
--------	---

PFIX

a. Functions - This subroutine adjusts the modulus of a periodic J coordinate.

b. Input -

P1	Hour angle measured eastward from noon (radians)
P2	Hour angle array at position J (radians)
P3	Hour angle array at position J+1 (radians)

c. Processing - No other subroutines are called.

This subroutine adjusts the modulus of the periodic input parameters to match P1 so that they can be used to interpolate data.

d. Output -

P4	Adjusted hour angle array at position J (radians) (corresponds to P2)
P5	Adjusted hour angle array at position J+1 (radians) (corresponds to P3)

INITAL

a. Functions - This subroutine computes the initial plasma distribution.

b. Input -

LL	Time label
LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
IEDIM	Number of particle species to be traced
ITMDIM	Maximum number of major time steps for this run
IRDIM	R dimension of FLXMAT arrays
NRGDIM	Energy dimension of FLXMAT arrays
KPDIM	KP dimension of FLXMAT arrays
ISPDIM	Plasma mass species dimension
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}(\text{eV})$)
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}(\#/\text{cm}^2\text{-s-ster-eV})$)
BNDLOC	Location of outer boundary of detailed particle traces
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VM	$(\text{Flux tube volume})^{-2/3} ((\text{Re}/nT)^{-2/3})$
ETA	Flux tube content array (weber^{-1})
ALAM	Energy invariant vector ($\text{eV}(\text{Re}/nT)^{2/3}$)
ALMDEL	Width of energy invariant channels
FKP	Kp at present time step
IEMAX	Maximum number of energy channels
AZERO	Statistical model weighting parameter
BZERO	Old results weighting parameter
CZERO	Traceback weighting paramater

c. Processing - following are the subroutines called and their major functions:

FLXVAL Function subprogram to calculate Kp-dependent flux value a $R=RR$ for particles of energy ENRG and $Kp=FKP$ by interpolating FLXMAT array

This subroutine can compute the initial plasma distribution based on a combination of empirical (Kp-based) data and results from previous runs. In practice, previous run results are used, if

available, and the empirical model is used otherwise. The rationale for this subroutine is discussed in Section 2.5.4 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

ETABEG Initial flux tube content distribution each time step (weber^{-1})

TIMINC

a. Function - This subroutine increments the time for the next electric and magnetic field record.

b. Input -

ITIME	Current time (year, day, seconds)
IINC	Increment time (year,day,seconds)

c. Processing - No other subroutines are called.

This subroutine increments the time and checks that it is within bounds.

d. Output -

ITMNEW	Time for next record (year,day,seconds)
--------	---

SETREF

a. Functions - This subroutine sets an upper limit reference flux.

b. Input -

LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
IEDIM	Number of particle species to be traced
ITMDIM	Maximum number of major time steps for this run
IRDIM	R dimension of FLXMAT arrays
NRGDIM	Energy dimension of FLXMAT arrays
KPDIM	KP dimension of FLXMAT arrays
ISPDIM	Plasma mass species dimension
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}(\text{eV})$)
FLXKP	KP values for which FLXMAT arrays are calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}(\#/\text{cm}^2\text{-s-ster-eV})$)
BNDLOC	Location of outer boundary of detailed particle traces
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VM	Flux tube volume $((\text{Re}/nT)^{-2/3})$
ETA	Flux tube content array (weber^{-1})
ALAM	Energy invariant vector ($\text{eV}(\text{Re}/nT)^{2/3}$)
ALMDEL	Width of energy invariant channels
IEMAX	Maximum number of energy channels
TIMTAG	Vector giving times at which E and B parameters are calculated (seconds)
TVAL	Current time (seconds)
ITMMAX	Maximum number of time labels

c. Processing - following are the subroutines called and their major functions:

TNORML	Utility function subprogram to calculate run-normalized time
FLXVAL	Function subprogram to calculate Kp-dependent flux value at $R=RR$ for particles of energy ENRG and $Kp=FKP$ by interpolating FLXMAT array

The rationale for this subroutine is discussed in Section 2.5.6 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

FLXREF Upper limit to reference flux ($\log_{10}(\#/cm^2\text{-s-ster-eV})$)

EFLOC

a. Functions - This subroutine finds the grid location (BI,BJ) of physical location (RVAL,XLT).

b. Input -

RVAL	Radial distance (Re)
XLT	Hour angle measured eastward from noon (radians)
LL	Time index
LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
ITMDIM	Maximum number of major time steps for this run
ALOCT	Grid local time array (radians eastward from noon)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)

c. Processing - following are the subroutines called and their major functions:

PFIX	Subroutine to adjust modulus of periodic J coordinate
------	---

This subroutine locate a point within a grid square and then linearly interpolates in I and J to give the non-integer locations.

d. Output -

BI	Non-integer I location of (RVAL,XLT)
BJ	Non-integer J location of (RVAL,XLT)

EFBNDY

a. Functions - This subroutine calculates the latitudes and widths of the electric field boundarys 2 and 3.

b. Input -

LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
ITMDIM	Maximum number of major time steps for this run
LL	Time label
EQEDG	Equatorward edge of auroral oval (degrees)
STDOFF	Standoff distance (Re)
VDROP	Cross-polar cap potential drop
ALOCT	Grid local time array (radians eastward from noon)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
COLAT	Grid colatitude array (radians)

c. Processing - following are the subroutines called and their major functions.

EFLOC	Subroutine to find grid location (BI,BJ) of physical location (RVAL,XLT)
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

The rationale for this subroutine is discussed in Section 2.4.2 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

A	Radius of ellipse measured in x (sunward) direction
B	Radius of ellipse measured in y (duskward) direction
DX	Offset of ellipse in x(duskward) direction
DY	Offset of ellipse in y (duskward) direction

FECON

a. Functions - This subroutine obtains the time-normalized observational values.

b. Input -

ISTART	Start time (year,day, seconds)
IINC	Increment time (year,day,seconds)
IEND	Stop time (year,day,seconds)
ITMDIM	Maximum number of major time steps per run of the program
NAUGEL	Number of elements in the augmented input array
ITMMAX	Number of time labels used
TIMTAG	Vector giving times at which E and B parameters are calculated

c. Processing - following are the subroutines called and their major functions:

PARGEN	Subroutine to obtain values from the environmental data base and interpolate or extrapolate as appropriate to provide data at a normalized time
TIMINC	Subroutine to increment time for next electric- and magnetic field record
TILT	Subroutine to calculate the Earth's tilt angle
DSTDFL	Subroutine to calculate default (Kp driven) Dst value
EQTDFL	Subroutine to calculate default (Kp driven) equatorward edge of auroral zone value
STNDFL	Subroutine to calculate the standoff distance
PCPDFL	Subroutine to calculate default (Kp driven) cross-polar-cap potential value
PATDFL	Subroutine to return polar-cap convection pattern typ
CLPDFL	Subroutine to determine whether or not to use the collapsed tail version of the magnetic field model

This subroutine calls Subroutine PARGEN to obtain time-normalized observational data that is then used to calculate other values needed by the electric field and magnetic field routines.

d. Output -

AUGPAR	Augmented data array for input values
MODE	Logical vector which gives whether each variable comes from actual data or a front end model.

PARGEN

a. Functions - Subroutine to obtain values from the environmental data base and interpolate or extrapolate as appropriate to provide data at a normalized time.

b. Input -

ISTART	Start time (year,day, seconds)
IINC	Increment time (year,day,seconds)
IEND	Stop time (year,day,seconds)
ITMDIM	Maximum number of major time steps per run of the program
NAUGEL	Number of elements in the augmented input array
ITMMAX	Maximum number of time labels
TIMTAG	Vector giving times at which E and B parameters are calculated
NELTS	Number of elements in output data array

c. Processing - following are the subroutines called and their major functions:

INDATA	Subroutine to read in data for the MSM during development and testing.
SMOOTH	Subroutine to extract data from input array.
DTXIPT	Subroutine to return interpolated polar cap patterns
DTNTRP	Subroutine to return interpolated data values
TIMINC	Subroutine to increment time for next electric and magnetic field record

This subroutine retrieves observational data from the environmental data base and interpolates as necessary to have input data for the entire run. Mode is set to TRUE if there exists data for the run within the time constraints of array IGAP. If the gap in the data is too long, MODE is set false and the KP-based default models are used to compute the input data needed.

d. Output -

PARRAY	Data array of interpolated data values
MODE	Logical variable telling which input parameters we have values for

INDATA

a. Functions - Subroutine to read in data for the MSM during development and testing.

b. Input -

PARAM	File name to read input data from (character)
STARTT	Start time (year,day,seconds)
ENDT	End time (year,day,seconds)
NDIM	Maximum number of data points to read

c. Processing - No other routines are called.

This routine opens a data file and reads the observation data. This routine will be replaced in the operational MSM with an interface routine to the AWS environmental data base.

d. Output -

DARRY	Array containing observational data
NUMNUM	Number of data points read

GETMAT

a. Functions - Subroutine to find the magnetic field matrices needed for interpolation in order to represent current geophysical conditions.

b. Input -

IWANT	Different magnetic field parameters = 1 - XMIN (GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 2 - YMIN (GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 3 - ZMIN (GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 4 - BMIN (Equatorial magnetic field strength (nT)) = 5 - VM (Flux tube volume (weber ⁻¹))
LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
BFPAR	Array containing geophysical parameters to use to find magnetic field matrices 1 = Standoff distance (Re) 2 = Tilt angle (degrees) 3 = Equatorward edge of auroral oval (degrees) 4 = Dst (nT) 5 = Collapse parameter

c. Processing - following are the subroutines called and their major functions:

FNDBRK	Subroutine to find the proper magnetic field indices that bracket the magnetic field input parameters
LOADBM	Subroutine to load individual magnetic field matrices from the magnetic field super matrix into the work magnetic field matrices
ZEROBM	Subroutine to zero out an individual magnetic field matrix within the working magnetic field matrices
OUTP	Utility subroutine to print out array information

d. Output -

WORK	Internal working array
BFLIM	Indices of magnetic field models read
BFEXST	Array of logical variables of which magnetic field model matrices exist

FNDBRK

a. Functions - Subroutine to find the magnetic field array indices that bracket the magnetic field input parameters.

b. Input -

PARVAL	Magnetic field geophysical parameter
PVALS	Array of magnetic field parameters
IPDIM	Dimension of PVALS

c. Processing - No other subroutines are called.

This routine finds the indices of the values in array PVAL that bracket the input geophysical parameter.

d. Output -

MIN	Lower array bound index
MAX	Upper array bound index

MEXIST

a. Functions - Logical function that checks to see if a particular B-matrix exists in the B supermatrix

b. Input -

MSTND	Number of standoff parameters, currently set to 5
MTILT	Number of tilt angle parameters, currently set to 5
MINED	Number of auroral equatorward edge parameters, currently set to 16
MDST	Number of Dst parameters, currently set to 8
MSTCH	Number of tail collapse parameters, currently set to 2

c. Processing - No other subroutines are called.

This function uses the FORTRAN "INQUIRE" command to check for the existence of needed B-field arrays/

d. Output -

MEXIST true if B field arrays exist, false otherwise

ZEROBM

a. Functions - Subroutine to zero out an individual magnetic field matrix within the working magnetic field matrices if the matrix does not exist in the supermatrix.

b. Input -

LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
I,J,K,L,M	Indices of working magnetic field matrices
WORK	Working magnetic field matrix

c. Processing - No other subroutines are called.

So that the program does not crash if a magnetic field matrix does not exist, it is zeroed out. The printout within the program prints that no magnetic field matrix existed for those geophysical conditions.

d. Output -

WORK	Zeroed out working magnetic field matrix
------	--

LOADBM

a. Processing - Subroutine to load individual magnetic field matrices from the magnetic field super-matrix into the work magnetic field matrices.

c. Input -

IWANT	5 different magnetic field parameters = 1 - XMIN (GSM X location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 2 - YMIN (GSM Y location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 3 - ZMIN (GSM Z location of where field line going through grid pt crosses the equatorial (B-field minimum) plane (Re)) = 4 - BMIN (Equatorial magnetic field strength (nT)) = 5 - VM ((Flux tube volume) ^{-2/3} ((Re/nT) ^{-2/3}))
LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
I,J,K,L,M	Indices of working magnetic field matrices
BFNDX	Array containing the indices of the magnetic field matrices to load for geophysical conditions of standoff distance, tilt angle, equatorward edge of the auroral oval, Dst, and tail collapse index

c. Processing - No other subroutines are called.

This subroutine loads individual magnetic field matrices from the offline magnetic field super-matrix into the work magnetic field matrices. A check is made to verify that the correct matrix has been retrieved. If this check fails the program will stop.

d. Output -

Work	Magnetic field matrix work array
------	----------------------------------

TCONV2

a. Functions - Function to take a start year, start decimal day, and current decimal day to produce a value that is the number of seconds from midnight of the start year and day.

b. Input -

YR	Current year
DD	Current day
STRTYR	Start year
STRTDD	Start day

c. Processing - No other subroutines are called.

d. Output -

TCONV2	Seconds from midnight of start day
--------	------------------------------------

TCONV3

a. Functions - Function to take a start year, start day, current year, current day, and current seconds of day to produce a value that is the number of seconds from midnight of the start year and day.

b. Input -

ITIME	Current time (year,day,seconds)
ISTART	Start time (year,day,seconds)

c. Processing - No other subroutines are called.

d. Output -

TCONV3	Seconds from the start year and day
--------	-------------------------------------

SMOOTH

a. Functions - Subroutine to extract and interpolate data from observational data into a usable form for the MSM.

b. Input -

DARRY	Array containing input observational data
NDIM	Maximum number of data points in DARRY
NUMNUM	Number of data points in DARRY
STARTY	Start year
STARTD	Start day

c. Processing - following are the subroutines called and their major functions:

TCONV2	Function to take a start year, start decimal day, and current decimal day to produce a value that is the number of seconds from midnight of the start year and day.
--------	---

This subroutine extracts the data relevant to the total run time.

d. Output -

NA	Total number of data points found
XA	Time (seconds)
YA	Data point
IERR	Error code
	= 0 no errors detected
	= -1 error in data

DTNTRP

a. Functions - Subroutine to return interpolated data.

b. Input -

X	Time (seconds)
NA	Number of data points
XA	Time array of observational data (seconds)
YA	Array of observed data

c. Processing - No other subroutines are called.

This subroutine takes the given time, finds the bracketing time in the XA array and linearly interpolates the observational data. It also does the differential DY/DX.

d. Output -

Y	Interpolated observational data
DYDX	Differential DY/DX
DELTA	The difference between the given time and the time of the observational data
IERR	Error code
	= 0 no problems in interpolation
	= -1 error detected

STNDOF

a. Functions - Subroutine to calculate the standoff distance from solar wind velocity and density data.

b. Input -

VELOC	Solar wind velocity data (km/sec)
DENS	Solar wind density data (cm^{-3})

c. Processing - No other subroutines are called.

This is based on the analytic calculation of Alpbach (1979).

d. Output -

STAND	Standoff distance (R_e)
-------	-----------------------------

FLXNIT

a. Functions - Subroutine to calculate Kp based particle number flux as function of R, energy and Kp.

b. Input -

IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
ISPDIM	Number of plasma mass species (electrons, H ⁺ , O ⁺ , etc.)
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated (log ₁₀ (eV))
FLXKP	KP values for which FLXMAT arrays are calculated

c. Processing - following are the subroutines called and their major functions:

FLXCAL	Subroutine to compute Kp-dependent flux at L=3, 4, 6.6, and 13 for electrons of given energy for a given Kp condition
THRCAL	Subroutine to calculate threshold flux at R=3, 4, 6.6, and 13 Re for given Kp and energy

This subroutine sets up the threshold and default fluxes for the program run.

d. Output -

FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values (log ₁₀ (#/cm ² -s-ster-eV))
THRMAT	Array of empirical fluxes as function of R, energy, and KP for use as threshold fluxes. (log ₁₀ (#/cm ² -s-ster-eV))

FLXCAL

a. Functions - Subroutine to compute Kp-dependent flux at L=3, 4, 6.6 and 13 for different species of given energy for a given Kp condition.

b. Input -

KPDIM	Number of Kp values for which FLXMAT is calculated
ISPDIM	Number of plasma mass species (electrons, H ⁺ , O ⁺ , etc.)
FNRLGL	One energy value for which FLXMAT arrays are calculated (log ₁₀ (eV))
FLKP	One Kp value for which FLXMAT arrays are calculated
ISPP	Chemical species identifier
	1 means electrons
	2 means H ⁺ ions
	3 means O ⁺ ions

c. Processing - No other subroutines are called.

The rationale for this subroutine is discussed in Section 2.5 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

FLX3	Flux at L=3 (#/cm ² -sec-ster-eV) for given energy for given Kp
FLX4	Flux at L=4 (#/cm ² -sec-ster-eV) for given energy for given Kp
FLX6	Flux at L=6 (#/cm ² -sec-ster-eV) for given energy for given Kp
FLX13	Flux at L=13 (#/cm ² -sec-ster-eV) for given energy for given Kp

FLXTRP

a. Functions - Function subprogram to return Log_{10} of empirical flux value interpolated between values at $L=3, 4, 6.6,$ and 13 .

b. Input -

RLOG	Log_{10} of radial distance to be interpolated
FLX3	Flux at $L=3$ ($\#/\text{cm}^2\text{-sec-ster-eV}$) for given energy for given K_p
FLX4	Flux at $L=4$ ($\#/\text{cm}^2\text{-sec-ster-eV}$) for given energy for given K_p
FLX6	Flux at $L=6$ ($\#/\text{cm}^2\text{-sec-ster-eV}$) for given energy for given K_p
FLX13	Flux at $L=13$ ($\#/\text{cm}^2\text{-sec-ster-eV}$) for given energy for given K_p

c. Processing - No other subroutines are called.

d. Output -

FLXTRP	Flux value interpolated in radial distance
--------	--

FLXVAL

a. Functions - Function subprogram to calculate Kp-dependent flux value for particles at a given radial distance, energy, and Kp conditions by interpolating the FLXMAT array.

b. Input -

ISP	Vector of chemical species identifier 1 means electrons 2 means H ⁺ ions 3 means O ⁺ ions
FKP	Kp for current time
RR	Radial distance of particle (Re)
PP	Hour angle (measured eastward from noon in radians) in equatorial plane
ENRG	Energy (eV)
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
ISPDIM	Number of plasma mass species (electrons, H ⁺ , O ⁺ , etc.)
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated (log ₁₀ (eV))
FLXKP	KP values for which FLXMAT arrays are calculated

c. Processing - following are the subroutines called and their major functions:

G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
--------	--

This subroutine performs a general 3-d interpolation in radial distance, Kp, and energy.

d. Output -

FLXVAL	Interpolated flux
--------	-------------------

EFIELD

a. Functions - Subroutine to return values of the potential on the northern hemisphere grid, southern hemisphere grid, and the average of the two hemispheres.

b. Input -

KPATT	Polar cap pattern type = 0 $B_z > 0$ = 1 $B_z < 0$ with stronger flow on dawn side of polar cap = 2 $B_z < 0$ with symmetric flow in polar cap = 3 $B_z < 0$ with stronger flow on dusk side of polar cap
PCP	Cross polar cap potential drop (kV)
DEQDT	Time rate of change of the equatorward edge of the auroral zone (degrees/hour)
ASOUTH	Southern hemisphere radius of ellipse measured in x (sunward) direction
BSOUTH	Southern hemisphere radius of ellipse measured in y (duskward) direction
DXS	Southern hemisphere offset of ellipse measured in x (sunward) direction
DYS	Southern hemisphere offset of ellipse measured in y (duskward) direction
ANORTH	Northern hemisphere radius of ellipse measured in x (sunward) direction
BNORTH	Northern hemisphere radius of ellipse measured in y (duskward) direction
DXN	Northern hemisphere offset of ellipse measured in x (sunward) direction
DYN	Northern hemisphere offset of ellipse measured in y (duskward) direction
COLAT	Grid colatitude array (radians)
ALOC1	Grid local time array (radians eastward from noon)
MODE	Logical variable telling which input parameters we have values for
LATLIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
JWRAP	Number of points of overlap in J direction
ITMCUR	Current time label
ITMDIM	Maximum number of major time steps per run of the program
ICNTRL	= 1 means different formulas are used in regions 1, 2, 3, = 2 means Heppner-Maynard formula is used for all latitudes, but scaled to externally specified cross polar cap potential drop and ellipse parameters

ITOP = -1 means Heppner-Maynard formula is used for all latitudes, unscaled.
 = -2 should not be used. It is the form used by the Rice Convection Model
 = 1 means the ellipse parameters A, B, DX, and DY for the polar cap boundary are computed internally in the subroutine. This is the normal mode for the MSM.

c. Processing - following are the subroutines called and their major functions:

EMODEL Returns the electric potential arrays on the northern hemisphere grid, southern hemisphere grid, and the average of the two hemispheres

The major function of this subroutine is to translate the KPATT pattern types given by the University of Texas, Dallas to the internally used IPATT pattern types of the Heppner-Maynard empirical model. It also keeps track of which hemisphere is being modelled and averages the output from the two hemispheres.

d. Output -

V Electric potential distribution on grid (average of VNORTH and VSOUTH) (Volts)
 VNORTH Northern hemisphere electric potential distribution (Volts)
 VSOUTH Southern hemisphere electric potential distribution (Volts)

PWRCAL

a. Functions - Subroutine to estimate precipitating electron energy flux poleward of the main MSM modeling region by comparing flux within the modeling region with AFGL statistical values.

b. Input -

LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
ISPDIM	Number of plasma mass species (electrons, H ⁺ , O ⁺ , etc.)
ITMCUR	Current time label
ITMDIM	Maximum number of major time steps per run of the program
FKP	Kp for current time
COLAT	Grid colatitude array (radians)
ALOC	Grid local time array (radians eastward from noon)
ALPHA	Latitudinal grid spacing vector
BETA	Longitudinal grid spacing vector
BNDLOC	Location of outer boundary of detailed particle traces
A	Radius of ellipse measured in x (sunward) direction
B	Radius of ellipse measured in y (duskward) direction
DX	Offset of ellipse measured in x (sunward) direction
DY	Radius of ellipse measured in y (duskward) direction

c. Processing - following are the subroutines called and their major functions:

THET	Function that gives equation for and electric field boundary ellipse in flat polar coordinates
------	--

The rationale for this subroutine is discussed in Section 2.6.5.2 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

FLXSUM	Total precipitation energy flux array (integrated over species for (1) electrons (2) H ⁺ and (3) O ⁺) (ergs/cm ² -sec)
EAVG	Average precipitating electron energy for (1) electrons, (2) H ⁺ and (3) O ⁺ (eV)

WKRATE

a. Functions - Function subroutine to evaluate weak precipitation loss rate.

b. Input -

RMID	R value at point where function is to be evaluated (R_e)
RPPMID	Estimate of plasmopause radius at midnight local time (R_e)
PMID	Local time hour angle in equatorial plane (radians from local noon)
ENRG	Particle energy (eV)

c. Processing - following are the subroutines called and their major functions:

RLYONS Function to evaluate Lyons-based weak loss rate

The rationale for this subroutine is discussed in Section 2.6.3 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

WKRATE Estimate of weak precipitation loss rate

RLYONS

a. Functions - Function to evaluate Lyons-based weak loss rate.

b. Input -

RMID	R value at point where function is to be evaluated (Re)
ENRG	Particle energy (eV)

c. Processing - No other subroutines are called.

The rationale for this subroutine is discussed in Section 2.6.3 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

RLYONS Estimate of weak precipitation loss rate from L. Lyons formula

G3TRPA

a. Functions - Utility function subprogram to perform a general 3-d linear interpolation of an angular array A(I,J,K) at point (BI,BJ,BK)

b. Input -

A	Array to be interpolated
IMAX	I dimension of array A
JMAX	J dimension of array A
KMAX	K dimension of array A
BI	Floating point value to interpolate in I dimension
BJ	Floating point value to interpolate in J dimension
BK	Floating point value to interpolate in K dimension

c. Processing - No other subroutines are called.

This subroutine uses the standard linear interpolation technique to interpolate an angular array in 3 dimensions.

d. Output -

G3TRPA Interpolated point from array A

THRCAL

a. Functions - Subroutine to calculate threshold flux at $R=3, 4, 6.6$, and $13 R_e$ for given K_p value and energy.

b. Input -

FNRGLG	One energy value for which FLXMAT arrays are calculated ($\log_{10}(\text{eV})$)
FKP	One K_p value for which FLXMAT arrays are calculated
NRGNDX	Energy calculation index
KPNDX	K_p calculation index
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of K_p values for which FLXMAT is calculated
FLXMAT	Array of empirical fluxes as function of R , energy, and K_p for use in calculating initial and boundary flux values ($\log_{10}(\#/\text{cm}^2\text{-s-ster-eV})$)

c. Processing - No other subroutines are called.

The rationale for this subroutine is discussed in Section 2.5.8 of the Final Report for Contract #F19628-87-K-0001.

d. Output -

THR	Value for threshold flux ($\text{particles}/\text{cm}^2\text{-s-ster-eV}$)
-----	--

FI.XDFL

a. Functions - Subroutine to calculate the energy-dependent ETA array from Kp and the location of the grid points by interpolating the empirical FLXMAT array.

b. Input -

ITM	Current time label
FKP	Current Kp
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (Re)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
VM	Flux tube volume $((Re/nT)^{-2/3})$
ALAM	Energy invariant vector $(eV(Re/nT)^{2/3})$
IRDIM	Number of R values for which the empirical flux array, FLXMAT, is calculated
NRGDIM	Number of preset energy values for which FLXMAT is calculated
KPDIM	Number of KP values for which FLXMAT is calculated
FLXMAT	Array of empirical fluxes as function of R, energy, and KP for use in calculating initial and boundary flux values ($\log_{10}(\#/cm^2-s-ster-eV)$)
FLXR	R values at which FLXMAT arrays are calculated (Re)
FLXNRG	Energy values for which FLXMAT arrays are calculated ($\log_{10}(eV)$)
FLXKP	KP values for which FLXMAT arrays are calculated
ALMDEL	Width of energy invariant channels
LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
ITMDIM	Maximum number of major time steps per run of the program
BNDLOC	Location of outer boundary of detailed particle traces
IEMAX	Maximum number of energy channels
IEDIM	Number of invariant energy species traced by the program
IFLAV	Chemical species identifier 1 means electrons 2 means H ⁺ ions 3 means O ⁺ ions
IEBEG	Beginning energy index
IEEND	Ending energy index

c. Processing - following are the subroutines called and their major functions:

FLXVAL	Function subprogram to calculate Kp-dependent flux value at given radial distance, energy and Kp by interpolating FLXMAT array.
--------	---

This routine is used as the MSM default model when full particle traces are done. The subroutine interpolates the FLXMAT arrays for the given energies for all grid points.

d. Output -

ETA	Flux tube content array (weber ⁻¹)
EFLUX	Precipitation energy flux array (ergs/cm ² -sec)

RMVBSH

a. Functions - Subroutine to extrapolate magnetic field matrices into extremely stretched or open regions.

b. Input -

WORK	Working magnetic field matrix
LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces

c. Processing - No other subroutines are called.

To prepare the magnetic field matrices for interpolation, previously assigned 'very large numbers' to signify that the magnetic field extended beyond 50 Re are removed. The subroutine looks for the closest value in local time (J) and linearly interpolates for the latitudinal (I) grid points.

d. Output -

WORK	Revised working magnetic field matrix
------	---------------------------------------

DSTDFL

a. Functions - Subroutine to calculate default (Kp-driven) Dst value

b. Input -

MODE	Logical variable telling which input parameters we have values for
AUGPAR	Augmented data array for input values
NAUGEL	Number of elements in the augmented input array
ITMDIM	Maximum number of major time steps per run of the program
ITM	<i>Current time label</i>
ITMMAX	Maximum number of time labels
TIMTAG	Vector giving times at which E and B parameters are calculated

c. Processing - following are the subroutines called and their major functions:

DTNTRP Subroutine to return interpolated data values

The subroutine uses the Kp value in the AUGPAR array for the given time and calculates Dst. This value is then put into the AUGPAR array. Subroutine DTNTRP is called to give the DY/DX value also needed in AUGPAR.

d. Output -

AUGPAR Augmented data array for input values

EQTDFL

a. Function - Subroutine to calculate default (Kp-driven) equatorward edge of the auroral oval value.

b. Input -

MODE	Logical variable telling which input parameters we have values for
AUGPAR	Augmented data array for input values
NAUGEL	Number of elements in the augmented input array
ITMDIM	Maximum number of major time steps per run of the program
ITM	Current time label
ITMMAX	Maximum number of time labels
TIMTAG	Vector giving times at which E and B parameters are calculated

c. Processing - following are the subroutines called and their major functions:

DTNTRP Subroutine to return interpolated data values

The subroutine uses the Kp value in the AUGPAR array for the given time and calculates the location of the equatorward edge of the auroral oval. The algorithm is given in a JGR paper by Gussenhoven et al. (1983). This value is then put into the AUGPAR array. Subroutine DTNTRP is called to give the DY/DX value also needed in AUGPAR.

d. Output -

AUGPAR Augmented data array for input values

PDPDFL

a. Functions - Subroutine to calculate default (Kp-driven) cross-polar-cap potential value.

b. Input -

MODE	Logical variable telling which input parameters we have values for
XKP	Current KP value

c. Processing - No other subroutines are called.

This subroutine uses the algorithms of Reiff et al to calculate the cross-polar-cap potential from given Kp.

d. Output -

PCP	Cross-polar cap potential (kV)
-----	--------------------------------

PATDFL

a. Functions - Subroutine to return polar cap convection pattern type of the basis of the measurements of the By and Bz components of the interplanetary magnetic field.

b. Input -

MODE	Logical variable telling which input parameters we have values for
MODEBY	Logical variable telling if we have By data
MODEBZ	Logical variable telling if we have Bz data

c. Processing - No other subroutines are called.

This subroutine determines the pattern type using interplanetary magnetic field data:

Polar cap pattern type

= 0 $B_z > 0$

= 1 $B_z < 0$ with stronger flow on dawn side of polar cap

= 2 $B_z < 0$ with symmetric flow in polar cap

= 3 $B_z < 0$ with stronger flow on dusk side of polar cap

If no data is available, pattern #2 is the default.

d. Output -

IPATT	Polar cap pattern type
-------	------------------------

CLPDFL

a. Functions - Subroutine to determine whether or not to use the collapsed tail version of the magnetic field model.

b. Input -

MODE Logical variable telling which input parameters we have values for

c. Processing - No other subroutines are called.

If mode is false, no value has been returned from the environmental data base and the default value is set to 1 (no collapse).

d. Output -

CLAPSE Value of magnetotail collapse parameter

STNDFL

a. Functions - Subroutine to calculate the standoff distance.

b. Input -

MODE	Logical variable telling which input parameters we have values for
MODVEL	Logical variable telling if we have solar wind velocity data
MODDEN	Logical variable telling if we have solar wind density data
SWVEL	Value of solar wind velocity data
SWDEN	Value of solar wind density data
XKP	Current Kp value

c. Processing - following are the subroutines called and their major functions:

STNDOF	Subroutine to calculate the standoff distance according to a formula published by Alpbach (1979)
--------	--

If solar wind velocity and density data are available Subroutine STNDOF does the actual calculation. Otherwise the default is a Kp-based formula.

d. Output -

STAND	Standoff distance (Re)
-------	------------------------

TILT

a. Functions - Subroutine to calculate the earth's tilt angle.

b. Input -

DAY	Day of year
YEAR	Year

c. Processing - No other subroutines are called.

This subroutine currently sets the tilt angle at 0.0.

d. Output -

XTILT	Tilt angle
-------	------------

AURL2S

a. Function - This subroutine determines the default Hardy precipitating ion flux.

b. Input -

FKP	Current Kp value
COLAT	Grid colatitude array (radians)
ALOCT	Time dependent magnetic local time grid
LATDIM	Number of latitudinal grid lines)
LTDIM	Number of local time (longitude) grid lines

c. Processing - following are the subroutines called and their major function:

REGEN	Subroutine used in evaluation of precipitatin ion fluxes
EFUN	Subroutine used in evaluation of precipitatin ion fluxes

These set of subroutines were developed by Dr. David Hardy at the Air Force Geophysics Laboratory, Bedford, Mass.

d. Output -

AEFLUX	The integral ion energy flux (ergs/cm ² -sec)
AEMEAN	The average energy (eV)

OPTERR

a. Functions - Subroutine to calculate the error and standard deviation between the MSM flux and observational geosynchronous satellite data.

b. Input -

FLUX	Particle flux array $(\text{cm}^2 \text{ s})^{-1}$
VM	$(\text{Flux tube volume})^{-2/3} ((R_e/nT)^{-2/3})$
LATDIM	Number of latitudinal grid spaces
LTDIM	Number of local time (longitudinal) grid spaces
IEDIM	Maximum number of energy channels
IEMAX	Number of invariant energy species traced by the program
ALAM	Energy invariant vector $(\text{eV}(R_e/nT)^{2/3})$
ALMDEL	Width of energy invariant channels
THRSH	Threshold energy beyond which particles are not traced and empirical data are used
ITMDIM	Maximum number of major time steps per run of the program
IFLAV	Chemical species identifier 1 means electrons 2 means H^+ ions 3 means O^+ ions
ISTART	Start time (year,day,seconds)
IEND	End time (year,day,seconds)
R	Array giving radial distance of grid pts in magnetospheric equatorial plane (R_e)
P	Array giving hour angle (measured eastward from noon in radians) of grid pts in equatorial plane
TIMTAG	Vector giving times at which E and B parameters are calculated
ITMMAX	Number of time labels in run

c. Processing - following are the subroutines called and their major functions:

TCONV2	Function to take a start year, start decimal day, and current decimal day to produce a value that is the number of seconds from midnight of the start year and day
READ3D	Subroutine to read a record from the standard MSM disk file format
TNORML	Utility function subprogram to calculate run-normalized time
EFLOC	Subroutine to find grid location (BI,BJ) of physical location (RVAL,XLT)
G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)

For the satellite times and energies available, the subroutine calculates and interpolates the model flux values at geosynchronous orbit. The error is defined as the average difference between the log of the model flux and the log of the satellite flux. The standard deviation is the usual definition. The subroutine averages over all available satellites. The format of the file 'EPSAT' will need to be changed when the MSM begins to access the AWS environmental data base. The standard deviation is not saved, just printed out.

d. Output -

ERSHFT Vector of error values

WRTVEC

a. Function - This subroutine writes to disk the energy invariant, the threshold energies, and the error shift for each particle energy species each time step.

b. Input -

NOUNIT	Logical unit number from which to read file
IRDOUT	Output direct access record number
IEMAX	Number of particle energy species for run
IEDIM	Dimension of particle energy species array
ALAM	Vector of energy invariant values used in run
THRSH	Vector of full-traceback/default value used in run
ERSHFT	Vector of error values for this time interval
IFLAV	Vector of chemical species identifier
	1 means electrons
	2 means H ⁺ ions
	3 means O ⁺ ions

c. Processing - No other subroutines are called.

This routine opens the appropriate file, writes the vectors to the file, and closes the file.

d. Output - The vectors are output to unit number NOUNIT.

CEXRAT

a. Function - Function subprogram to return charge exchange loss rate (sec^{-1}) for ions of species ISP, energy ENRG (eV), at L=RLOC (Re) for sunspot number SSN. This routine is based on a table generated by James Bishop of the University of Michigan.

b. Input -

ISP	Species Identifier ISP=2 for H ⁺ Ions ISP=3 for O ⁺ Ions
ENRG	Energy in eV
RLOC	Radial location (Re)
SSN	Sunspot number
DKTIME	Table of ion decay times
IRDK	Radial dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sunspot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array

c. Processing - following are the subroutines called and their major functions:

G3NTRP	Utility function subprogram to perform a general 3-d linear interpolation of an arbitrary array A(I,J,K) at point (BI,BJ,BK)
--------	--

This routine interpolates the decay time array in radial distance, energy and sunspot number.

d. Output -

CEXRAT	Ion charge exchange loss rate (sec^{-1})
--------	---

READDK

a. Function - Subroutine to read ion charge exchange decay table furnished by James Bishop of the University of Michigan.

b. Input -

IRDK	Radial dimension of DKTIME array
INRGDK	Energy dimension of DKTIME array
ISOLDK	Sunspot number dimension of DKTIME array
IONDK	Number of ion species in DKTIME array

c. Processing -

Reads the input table of ion decay times.

d. Output -

DKTIME	Table of ion decay times
--------	--------------------------

FLX2ET

a. Function - Convert from particle flux (particles/cm²/s) to invariant density eta (particles/Weber)

b. Input -

LATDIM	Number of latitudinal grid points
LTDIM	Number of longitudinal (local time) grid points
IEDIM	Maximum number of model energy channels
IEMAX	Actual number of model energy channels
ITMDIM	Maximum time depth
ITM	Time index
IFLAV	Vector giving charge and mass species of particles 1 = electrons, 2 = H ⁺ ions, 3 = O ⁺ ions
ALAM	Energy invariant vector
ALMDEL	Width of invariant energy channels
BNDLOC	Boundary location array
VM	(Flux tube volume) ^{-2/3}
FLUX	Particle number flux (#/cm ² /s)

c. Processing - No other subroutines are called.

This subroutine uses a simple formula to translate from number flux FLUX to invariant density ETA.

d. Output -

ETA	Invariant number density (particles / Weber)
-----	--

Appendix B

ET2FLX

a. Function - Convert from invariant density eta (particles/Weber) to particle number flux (particles/cm²/s)

b. Input -

LATDIM	Number of latitudinal grid points
LTDIM	Number of longitudinal (local time) grid points
IEDIM	Maximum number of model energy channels
IEMAX	Actual number of model energy channels
ITMDIM	Maximum time depth
ITM	Time index
IFLAV	Vector giving charge and mass species of particles 1 = electrons, 2 = H ⁺ ions, 3 = O ⁺ ions
ALAM	Energy invariant vector
ALMDEL	Width of invariant energy channels
BNDLOC	Boundary location array
VM	(Flux tube volume) ^{-2/3}
ETA	Invariant number density (particles / Weber)

c. Processing - No other subroutines are called.

This subroutine uses a simple formula to translate from invariant density ETA to number flux FLUX.

d. Output -

FLUX	Particle number flux (#/cm ² /s)
------	---

90/06/30
12:04:35

msml.com

1

```
$ define bfield [.bfields] /process
$ define for005 msmin.dat /process
$ define for006 print3.dat /process
$ define for009 error1.dat /process
$ define for010 efield.coeff /process
$ define for011 bfield.coeff /process
$ define for012 parray.dat /process
$ define for013 pieng.dat /process
$ define for015 print.dat /process
$ define for018 tt /process
$ define for020 dktable.dat /process
$ define for021 etabnd.dat /process
$ define for023 shift.dat /process
$ define for024 piflx.dat /process
$ define for025 eavg.dat /process
$ define for026 flxsum.dat /process
$ define for027 flux.dat /process
$ define for035 v.dat /process
$ define for036 vm.dat /process
$ define for037 bmin.dat /process
$ define for038 xmin.dat /process
$ define for039 ymin.dat /process
$ define for040 zmin.dat /process
$ define for041 aloct.dat /process
$ define for042 colat.dat /process
$ define for043 bndloc.dat /process
$ define for044 updat.dat /process
$ define for045 vnorth.dat /process
$ define for046 vsouth.dat /process
$ define efcoef efcoef.dat /process
$ define coord coord.dat /process
$ define hardy hardy.dat /process
$ define epsat epsat.dat /process
$ define epions epions.dat /process
$ define ioneng ioneng.dat /process
$ define ionnum ionnum.dat /process
$ define enchan enchan.dat /process
$ define fkp fkp.dat /process
$ define dst dst.dat /process
$ define eqedge eqedge.dat /process
$ define clapse clapse.dat /process
$ define swden swden.dat /process
$ define swvel swvel.dat /process
$ define pcp pcp.dat /process
$ define xipatt xipatt.dat /process
$ run msml
```

90/06/30
12:03:44

msm1.for

1

```
PROGRAM MSMCON
C
C VERSION 1.0
C 1.1
C 1.2
C 1.3
C
C PROGRAMMER: BOB SPIRO
C
C PURPOSE: MSMCON IS THE MAIN PROGRAM FOR THE MAGNETOSPHERIC
C SPECIFICATION MODEL. MSMCON CALLS THE SUBROUTINES PPTCON,
C HIEPAR, AND OUTPUT.
C
C PARAMETER (LATDIM=62,LTDIM=51,IEDIM=30,ITMDIM=50,
C 2 NAUGEL=28,IRDIM=4,NRGDIM=29,KPDIM=7,ISPDIM=3,
C 3 IRDK=18,INRGDK=13,ISOLDK=2,IONDK=2,KPPLUS=9)
C
C LATDIM GIVES THE NUMBER OF LATITUINAL GRID LINES (ALIASED AS IDIM
C IN SOME ROUTINES)
C LTDIM GIVES THE NUMBER OF LOCAL TIME (LONGITUDE) GRID LINES (ALIASED
C AS JDIM IN SOME ROUTINES)
C IEDIM GIVES THE NUMBER OF INVARIANT ENERGY SPECIES TRACED BY THE
C PROGRAM
C ITMDIM GIVES THE MAXIMUM NUMBER OF MAJOR TIME STEPS PER RUN OF THE
C PROGRAM
C NAUGEL GIVES THE NUMBER OF ELEMENTS IN THE AUGMENTED INPUT ARRAY
C
C IRDIM GIVES THE NUMBER OF R VALUES FOR WHICH THE EMPIRICAL FLUX
C ARRAY, FLXMAT, IS CALCULATED
C NRGDIM GIVES THE NUMBER OF PRESET ENERGY VALUES FOR WHICH FLXMAT IS
C CALCULATED
C KPDIM GIVES THE NUMBER OF KP VALUES FOR WHICH FLXMAT IS CALCULATED
C
C KPPLUS GIVES THE KP DIMENSION OF THE AUGMENTED FLXMAT ARRAY
C (KPDIM + 2)
C
C ISPDIM GIVES THE NUMBER OF PLASMA MASS SPECIES (ELECTRONS,H+,O+,ETC)
C
C CHARACTER*80 CHID
C
C COMMON LUNIT CONTAINS LOGICAL UNIT NUMBERS FOR INPUT AND OUTPUT
C COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
C 1 LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
C 2 LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
C 3 LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
C COMMON IRDREC CONTAINS RECORD NUMBERS FOR PARTICE TRACE (IRECPT)
C AND E AND B FIELD FILES (IRECEB)
C COMMON /IRDREC/ IRECPT,IRECEB,IRDBEG
C
C COMMON SUN CONTAINS THE SUN SPOT NUMBER FOR THE ION CHARGE
C EXCHANGE ALGORITHM
C COMMON /SUN/ SSN
C
C WITH FEW EXCEPTIONS, MOST OF THE ARRAYS USED IN THE PROGRAM ARE
C DIMENSIONED HERE IN MSMCON.
C
C DIMENSION ISTART(3),IINC(3),IEND(3)
C DIMENSION ETA(LATDIM,LTDIM,IEDIM),EFLUX(LATDIM,LTDIM,IEDIM),
C 2 FLXSUM(LATDIM,LTDIM,3),EAVG(LATDIM,LTDIM,3),
C 3 FLUX(LATDIM,LTDIM,IEDIM)
C
C DIMENSION VM(LATDIM,LTDIM,ITMDIM),V(LATDIM,LTDIM,ITMDIM),
```

MSM00010
MSM00020
MSM00030
MSM00040
MSM00050
MSM00060
MSM00070
MSM00080
MSM00090
MSM00100
MSM00110
MSM00120
MSM00130
MSM00140
MSM00150
MSM00160
MSM00170
MSM00180
MSM00190
MSM00200
MSM00210
MSM00220
MSM00230
MSM00240
MSM00250
MSM00260
MSM00270
MSM00280
MSM00290
MSM00300
MSM00310
MSM00320
MSM00330
MSM00340
MSM00350
MSM00360
MSM00370
MSM00380
MSM00390
MSM00400
MSM00410
MSM00420
MSM00430
MSM00440
MSM00450
MSM00460
MSM00470
MSM00480
MSM00490
MSM00500
MSM00510
MSM00520
MSM00530
MSM00540
MSM00550
MSM00560
MSM00570
MSM00580
MSM00590
MSM00600
MSM00610
MSM00620
MSM00630
MSM00640

90/06/30
12:03:44

2

msml.for

```
1  ETABEG (LATDIM, LTDIM, IEDIM), ETABND (LTDIM, IEDIM, ITMDIM),      MSM00650
2  FLXBND (LTDIM, IEDIM, ITMDIM),      MSM00660
3  BNDLOC (LTDIM, ITMDIM), ALAM (IEDIM), COLAT (LATDIM, LTDIM),      MSM00670
4  ALOCT (LATDIM, LTDIM), BMIN (LATDIM, LTDIM), XMIN (LATDIM, LTDIM),  MSM00680
5  YMIN (LATDIM, LTDIM), ZMIN (LATDIM, LTDIM), TIMTAG (ITMDIM),      MSM00690
6  ALMDEL (IEDIM)      MSM00700
C      MSM00710
      DIMENSION VNORTH (LATDIM, LTDIM), VSOUTH (LATDIM, LTDIM)      MSM00720
C      MSM00730
      DIMENSION IFLAV (IEDIM), BNDMAX (LTDIM),      MSM00740
1  ALPHA (LATDIM), BETA (LATDIM), BIR (LATDIM, LTDIM),      MSM00750
2  ASOUTH (3, ITMDIM), BSOUTH (3, ITMDIM), DXS (3, ITMDIM), DYS (3, ITMDIM),  MSM00760
3  ANORTH (3, ITMDIM), BNORTH (3, ITMDIM), DXN (3, ITMDIM), DYN (3, ITMDIM),  MSM00770
4  VMLOSS (LATDIM, LTDIM, ITMDIM),      MSM00780
5  TETA (LATDIM), PHI (LTDIM), SINI (LATDIM, LTDIM)      MSM00790
C      MSM00800
      DIMENSION R (LATDIM, LTDIM, ITMDIM), P (LATDIM, LTDIM, ITMDIM)      MSM00810
C      MSM00820
      DIMENSION SCTFRC (IEDIM, LTDIM, ITMDIM), RPP (ITMDIM)      MSM00830
C      MSM00840
      DIMENSION DKTIME (IRDK, INRGDK, ISOLDK, IONDK)      MSM00850
C      MSM00860
C      MSM00870
      DIMENSION ID (20), RID (20), THRSH (IEDIM), ERSHFT (IEDIM)      MSM00880
C      MSM00890
C      MSM00900
      DIMENSION AUGPAR (NAUGEL, ITMDIM)      MSM00910
      LOGICAL*1 MODE (NAUGEL, ITMDIM)      MSM00920
C      MSM00930
      DIMENSION FLXMAT (IRDIM, NRGDIM, KPPLUS, ISPDIM)      MSM00940
      DIMENSION THRMAT (IRDIM, NRGDIM, KPDIM)      MSM00950
      DIMENSION FLXR (IRDIM), FLXNRG (NRGDIM), FLXKP (KPDIM)      MSM00960
C      MSM00970
C      CALL PRECIPITATION AND PARTICLE TRACER CONTROL      MSM00980
C      MSM00990
C      MSM01000
C      INPUT MAJOR TIME STEP INCREMENT (YEARS, DAYS, SECONDS)      MSM01010
      IINC (1) = 0      MSM01020
      IINC (2) = 0      MSM01030
      IINC (3) = 900      MSM01040
C      MSM01050
C      READ IN START YEAR, START DAY, START TIME IN SECONDS      MSM01060
      READ (5, *) IYEAR, IDAY, ISEC      MSM01070
      ISTART (1) = IYEAR - (IYEAR/100) * 100      MSM01080
      ISTART (2) = IDAY      MSM01090
      ISTART (3) = ISEC      MSM01100
C      MSM01110
C      READ IN STOP YEAR, STOP DAY, STOP TIME IN SECONDS      MSM01120
      READ (5, *) IYEAR, IDAY, ISEC      MSM01130
      IEND (1) = IYEAR - (IYEAR/100) * 100      MSM01140
      IEND (2) = IDAY      MSM01150
      IEND (3) = ISEC      MSM01160
C      MSM01170
C      READ IN START RECORD NUMBER      MSM01180
      READ (5, *) IRECEB      MSM01190
      IRECPT = IRECEB      MSM01200
C      MSM01210
C      READ IN STRING TO IDENTIFY RUN      MSM01220
      READ (5, *) CHID      MSM01230
C      MSM01240
C      READ IN SUN SPOT NUMBER SSN      MSM01250
      READ (5, *) SSN      MSM01260
C      MSM01270
      WRITE (6, *) ' ISTART=', ISTART      MSM01280
```

90/06/30
12:03:44

msm1.for

3

```
WRITE(6,*) 'IEND=', IEND
WRITE(6,*) 'IRECPT=', IRECPT, 'IRECEB=', IRECEB
WRITE(6,*) 'CHID=', CHID
WRITE(6,*) 'SSN=', SSN

C
C      IRDBEG=IRECPT
C
C
C      INITIALIZE TIMER
C
C      CALLS TO DTIME ARE PLACED IN THE CODE TO PROVIDE TIMING INFORMATION
C      FOR DIFFERENT SECTIONS OF THE PROGRAM.  THE DUMMY SUBROUTINE STUB
C      DTIME NEEDS TO BE REPLACED WITH MACHINE-SPECIFIC TIMING CALLS IN
C      ORDER TO OBTAIN REAL TIMING INFORMATION.
C
C      CALL DTIME(ITODST)
C
C      CALL TO PPTCON READS AND ASSIMILATES AVAILABLE INPUT DATA, PRODUCES
C      E AND B FIELD MATRICES FOR PERIOD OF INTEREST, AND COMPUTES
C      PARTICLE FLUXES IN MAGNETOSPHERE AND PRECIPITATING INTO
C      IONOSPHERE.
C
C      CALL PPTCON(ISTART, IINC, IEND, LATDIM, LTDIM, IEDIM, ITMDIM,
1      IRDK, INRGDK, ISOLDK, IONDK,
1      IRDIM, NRGDIM, KPDIM, KPPLUS, ISPDIM, FLXR, FLXNRG, FLXKP, FLXMAT,
2      THRMAT, CHID, ID, RID, SCTFRG, IFLAV, BNDMAX, ALPHA, BETA, TETA, PHI, BIR,
3      SINI, TIMTAG, COLAT, ALOCT, V, VNORTH, VSOUTH, ASOUTH, BSOUTH, DXS, DYS,
4      ANORTH, BNORTH, DXN, DYN, VM, VMLOSS, BMIN, XMIN, YMIN, ZMIN,
5      ALAM, ETABEG, ETABND, FLXBND, BNDLOC, ETA, FLUX, EFLUX, FLXSUM,
6      EAVG, R, P,
7      ALMDEL, NAUGEL, AUGPAR, MODE, RPP, THRSH, ERSHT, DKTIME)

C
C      FIND TIME TO EXECUTE PPTCON
C
C      CALL DTIME(IPPTIM)
C      WRITE (LUERR,*) 'THE TIME SPENT IN PPTCON IS ',
1      ABS(IPPTIM-ITODST)/100., ' SECONDS.'

C      CALL HIEPAR
C      CALL TO HIEPAR GIVES THE STATISTICS-BASED HIGH ENERGY (> 300 KEV)
C      OUTPUT OF THE PROGRAM.  THE JULY 1, 1990, VERSION OF THE MSM FOR
C      DELIVERY TO THE AIR FORCE CONTAINS ONLY A DUMMY STUB FOR HIEPAR.
C
C      CALL HIEPAR
C
C      FIND TIME TO EXECUTE HIEPAR
C
C      CALL DTIME(IHIETM)
C      WRITE (LUERR,*) 'THE TIME SPENT IN HIEPAR IS ',
1      ABS(IHIETM-IPPTIM)/100., ' SECONDS.'

C      CALL OUTPUT
C      IN THE SFC OPERATIONAL VERSION OF THE MSM, THE CALL TO OUTPUT
C      WILL OUTPUT MSM RESULTS TO THE ENVIRONMENTAL DATA BASE.
C      THE JULY 1, 1990, DELIVERY VERSION OF THE MSM CONTAINS A STUB FOR
C      SUBROUTINE OUTPUT.
C
C      CALL OUTPUT
C
C      FIND TIME TO EXECUTE OUTPUT
C
C      CALL DTIME(IOUTIM)
C      WRITE (LUERR,*) 'THE TIME SPENT IN OUTPUT IS ',
1      ABS(IOUTIM-IHIETM)/100., ' SECONDS.'
```

MSM01290
MSM01300
MSM01310
MSM01320
MSM01330
MSM01340
MSM01350
MSM01360
MSM01370
MSM01380
MSM01390
MSM01400
MSM01410
MSM01420
MSM01430
MSM01440
MSM01450
MSM01460
MSM01470
MSM01480
MSM01490
MSM01500
MSM01510
MSM01520
MSM01530
MSM01540
MSM01550
MSM01560
MSM01570
MSM01580
MSM01590
MSM01600
MSM01610
MSM01620
MSM01630
MSM01640
MSM01650
MSM01660
MSM01670
MSM01680
MSM01690
MSM01700
MSM01710
MSM01720
MSM01730
MSM01740
MSM01750
MSM01760
MSM01770
MSM01780
MSM01790
MSM01800
MSM01810
MSM01820
MSM01830
MSM01840
MSM01850
MSM01860
MSM01870
MSM01880
MSM01890
MSM01900
MSM01910
MSM01920

90/06/30
12:03:44

msml.for

4

```
C
C      FIND THE TOTAL TIME OF THE RUN
C
C      CALL DTIME(ITODED)
C      WRITE (LUERR,*) 'THE TOTAL TIME SPENT IN MSMCON IS ',
C      1      ABS(ITODED-ITODST)/100.,' SECONDS.'
C
C      STOP
C      END
C
C ***** P P T C O N *****
C
C      SUBROUTINE PPTCON(ISTART,IINC,IEND,LATDIM,LTDIM,IEDIM,ITMDIM,
1      IRDK,INRGDK,ISOLDK,IONDK,
1      IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,
2      THRMAT,CHID,ID,RID,SCTFRC,IFLAV,BNDMAX,ALPHA,
3      BETA,TETA,PHI,BIR,SINI,TIMTAG,COLAT,
4      ALOCT,V,VNORTH,VSOUTH,ASOUTH,BSOUTH,DXS,DYS,
5      ANORTH,BNORTH,DXN,DYN,VM,VMLOSS,BMIN,XMIN,YMIN,ZMIN,
6      ALAM,ETABEG,ETABND,FLXBND,BNDLOC,ETA,FLUX,EFLUX,FLXSUM,EAVG,
7      R,P,
8      ALMDEL,NAUGEL,AUGPAR,MODE,RPP,THRSH,ERSHFT,DKTIME)
C
C      VERSION 1.1      DATE: NOVEMBER 23, 1988
C      1.2      SEPTEMBER 8, 1989
C      1.3      NOVEMBER 17, 1989
C      1.4 (APRIL, 1990 VERSION)      MARCH 26, 1990
C      1.5 (JULY 1, 1990 VERSION)      JULY 01, 1990
C
C      PROGRAMMER:  BRYAN BALES/R.W. SPIRO
C
C      PURPOSE:  THIS EXECUTIVE ROUTINE CONTROLS THE ELECTRIC AND MAGNETIC
C      FIELD MODEL (EBCON) AND THE RUNNING OF THE PARTICLE
C      TRACE ALGORITHM (PPTM).
C
C      CALLING PARAMETERS:
C      ISTART      START YEAR, DAY, AND TIME (SEC)
C      IINC      MAJOR TIME STEP INCREMENT (YEAR, DAY, TIME)
C      IEND      ENDING YEAR, DAY, AND TIME (SEC)
C      LATDIM      NUMBER OF LATITUDINAL GRID SPACES
C      LTDIM      NUMBER OF LOCAL TIME (LONGITUDINAL) GRID SPACES
C      IEDIM      NUMBER OF PARTICLE SPECIES TO BE TRACED
C      ITMDIM      MAXIMUM NUMBER OF MAJOR TIME STEPS FOR THIS RUN
C      IRDK      R DIMENSION OF DKTIME ARRAY
C      INRGDK      ENERGY DIMENSION OF DKTIME ARRAY
C      ISOLDK      SUN SPOT NUMBER DIMENSION OF DKTIME ARRAY
C      IONDK      NUMBER OF ION SPECIES IN DKTIME ARRAY
C      IRDIM      R DIMENSION OF FLXMAT ARRAYS
C      NRGDIM      ENERGY DIMENSION OF FLXMAT ARRAYS
C      KPDIM      NUMBER OF KP VALUES FOR FLXMAT ARRAYS
C      KPPLUS      AUGMENTED KP DIMENSION OF FLXMAT (KPDIM+2)
C      ISPDIM      PLASMA MASS SPECIES DIMENSION
C      FLXR      R VALUES AT WHICH FLXMAT ARRAYS ARE CALCULATED
C      FLXNRG      ENERGY VALUES FOR WHICH FLXMAT ARRAYS ARE CALCULATED
C      FLXKP      KP VALUES FOR WHICH FLXMAT ARRAYS ARE CALCULATED
C      FLXMAT      ARRAY OF EMPIRICAL FLUXES AS FUNCTION OF R, ENERGY,
C      AND KP FOR USE IN CALCULATING INITIAL AND BOUNDARY
C      FLUX VALUES.(COMPUTED IN CALL TO FLXNIT)
C      THRMAT      ARRAY OF EMPIRICAL FLUXES AS FUNCTION OF R, ENERGY,
C      AND KP FOR USE AS THRESHOLD FLUXES (COMPUTED IN CALL
C      TO FLXNIT)
C      CHID      CHARACTER HEADER STRING (UP TO 80 CHARACTERS)
C      ID      INTEGER HEADER VECTOR
C      ID(1) = YEAR
```

MSM01930
MSM01940
MSM01950
MSM01960
MSM01970
MSM01980
MSM01990
MSM02000
MSM02010
MSM02020
MSM02030
MSM02040
MSM02050
MSM02060
MSM02070
MSM02080
MSM02090
MSM02100
MSM02110
MSM02120
MSM02130
MSM02140
MSM02150
MSM02160
MSM02170
MSM02180
MSM02190
MSM02200
MSM02210
MSM02220
MSM02230
MSM02240
MSM02250
MSM02260
MSM02270
MSM02280
MSM02290
MSM02300
MSM02310
MSM02320
MSM02330
MSM02340
MSM02350
MSM02360
MSM02370
MSM02380
MSM02390
MSM02400
MSM02410
MSM02420
MSM02430
MSM02440
MSM02450
MSM02460
MSM02470
MSM02480
MSM02490
MSM02500
MSM02510
MSM02520
MSM02530
MSM02540
MSM02550
MSM02560

90/06/30
12:03:44

msm1.for

5

C	ID(2) = DAY	MSM02570
C	ID(3) = HOURS	MSM02580
C	ID(4) = MINUTES	MSM02590
C	ID(5) = SECONDS	MSM02600
C	ID(6) = PRESENTLY UNUSED	MSM02610
C	ID(7) = PRESENTLY UNUSED	MSM02620
C	ID(8) = FIRST DIMENSION OF OUTPUT ARRAY	MSM02630
C	ID(9) = SECOND DIMENSION OF OUTPUT ARRAY	MSM02640
C	ID(10) = THIRD DIMENSION OF OUTPUT ARRAY	MSM02650
C	ID(11) = TIME INDEX L	MSM02660
C	ID(12) - ID(20) = PRESENTLY UNUSED	MSM02670
C	RID	REAL HEADER VECTOR
C	RID(1) = TIMTAG	MSM02680
C	RID(2) = KP	MSM02690
C	RID(3) = POLAR CAP POTENTIAL DROP	MSM02700
C	RID(4) = TIME DERIVATIVE OF LOCATION OF	MSM02710
C	LOW-LATITUDE EDGE OF AURORAL PRECIPITATION	MSM02720
C	RID(5) - RID(20) = PRESENTLY UNUSED	MSM02730
C	SCTFRC	ASSUMED PITCH ANGLE SCATTERING EFFICIENCY
C	IFLAV	CHEMICAL SPECIES IDENTIFIER
C	1 MEANS ELECTRONS	MSM02740
C	2 MEANS H+ IONS	MSM02750
C	3 MEANS O+ IONS	MSM02760
C	BNDMAX	MAXIMUM I VALUE OF MODEL BOUNDARY ALONG EACH LOCAL
C	TIME GRID LINE DURING THE DURATION OF THIS RUN.	MSM02770
C	ALPHA	LATITUDINAL GRID SPACING VECTOR
C	BETA	LONGITUDINAL GRID SPACING VECTOR
C	TETA	COLATITUDE OF I GRID LINES (RADIAN) (USED ONLY IN
C	SUBROUTINE RDGRID)	MSM02780
C	PHI	NON-ROTATED HOUR ANGLE OF J GRID LINES (USED ONLY
C	IN SUBROUTINE RDGRID)	MSM02790
C	BIR	RADIAL COMPONENT OF IONOSPHERIC MAGNETIC FIELD (NT)
C	SINI	SIN(MAGNETIC FIELD INCLINATION ANGLE)
C	TIMTAG	VECTOR GIVING TIMES AT WHICH E AND B PARAMETERS ARE
C	CALCULATED	MSM02900
C	COLAT	GRID COLATITUDE ARRAY (RADIAN)
C	ALOC	GRID LOCAL TIME ARRAY (RADIAN EASTWARD FROM NOON)
C	V	ELECTRIC POTENTIAL DISTRIBUTION ON GRID (OUTPUT FROM
C	EBCON) (AVERAGE OF VNORTH AND VSOUTH)	MSM02910
C	VNORTH	NORTHERN HEMISPHERE ELECTRIC POTENTIAL DISTRIBUTION
C	(OUTPUT FROM EBCON)	MSM02920
C	VSOUTH	SOUTHERN HEMISPHERE ELECTRIC POTENTIAL DISTRIBUTION
C	(OUTPUT FROM EBCON)	MSM02930
C	ASOUTH, BSOUTH, DSX, DYS, ANORTH, BNORTH, DXN, DYN	MSM02940
C	SPECIFICATIONS OF ELECTRIC FIELD MODEL BOUNDARY	MSM02950
C	ELLIPSES	MSM02960
C	VM	FLUX TUBE VOLUME (OUTPUT FROM EBCON)
C	VMLOSS	GEOMETRIC FACTOR USED IN CALCULATING PRECIPITATION
C	LOSS	MSM03000
C	BMIN	EQUATORIAL MAGNETIC FIELD STRENGTH (NT)
C	XMIN	GSM X LOCATION OF WHERE FIELD LINE GOING THROUGH
C	GRID PT CROSSES THE EQUATORIAL (B-FIELD MINIMUM)	MSM03010
C	PLANE	MSM03020
C	YMIN	GSM Y LOCATION OF WHERE FIELD LINE GOING THROUGH
C	GRID PT CROSSES THE EQUATORIAL PLANE	MSM03030
C	ZMIN	GSM Z LOCATION OF WHERE FIELD LINE GOING THROUGH
C	GRID PT CROSSES THE (B-FIELD MINIMUM) PLANE	MSM03040
C	ALAM	ENERGY INVARIANT VECTOR
C	ETABEG	INITIAL FLUX TUBE CONTENT DISTRIBUTION EACH TIME
C	STEP	MSM03050
C	ETABND	BOUNDARY FLUX TUBE CONTENT DISTRIBUTION
C	FLXBND	FLUX VALUES AT BOUNDARY
C	BNDLOC	LOCATION OF OUTER BOUNDARY OF DETAILED PARTICLE
C	TRACES	MSM03060
		MSM03070
		MSM03080
		MSM03090
		MSM03100
		MSM03110
		MSM03120
		MSM03130
		MSM03140
		MSM03150
		MSM03160
		MSM03170
		MSM03180
		MSM03190
		MSM03200

90/06/30
12:03:44

6

msml.for

```
C      ETA      FLUX TUBE CONTENT ARRAY (PARTICLES/UNIT MAGNETIC      MSM03210
C                                     FLUX)      MSM03220
C      FLUX      PARTICLE FLUX AT GRID PTS (PARTICLES/CM**2/S)      MSM03230
C      EFLUX      PRECIPITATING ENERGY FLUX ARRAY      MSM03240
C      FLXSUM      TOTAL PRECIPITATION ENERGY FLUX ARRAY (INTEGRATED      MSM03250
C                                     OVER SPECIES) FOR (1) ELECTRONS (2) H+ & (3) O+      MSM03260
C      EAVG      AVERAGE PRECIPITATING ELECTRON ENERGY      MSM03270
C                                     FOR (1) ELECTRONS (2) H+ AND (3) O+      MSM03280
C      R      ARRAY GIVING RADIAL DISTANCE OF GRID PTS IN      MSM03290
C                                     MAGNETOSPHERIC EQUATORIAL PLANE (RE)      MSM03300
C      P      ARRAY GIVING HOUR ANGLE (MEASURED EASTWARD FROM      MSM03310
C                                     NOON IN RADIANS) OF GRID PTS IN EQUATORIAL PLANE      MSM03320
C      ALMDEL      WIDTH OF ENERGY INVARIANT CHANNELS      MSM03330
C      NAUGEL      DIMENSION OF INPUT ARRAYS      MSM03340
C      AUGPAR      AUGMENTED DATA ARRAY FOR INPUT VALUES      MSM03350
C      MODE      LOGICAL VARIABLE TELLING WHICH INPUT PARAMETERS WE      MSM03360
C                                     HAVE VALUES FOR      MSM03370
C      RPP      NOMINAL RADIUS OF PLASMAPAUSE DEFINED AS DIPOLE      MSM03380
C                                     MAPPING OF EQUATORWARD EDGE OF AURORAL ZONE TO      MSM03390
C                                     MAGNETOSPHERIC EQUATORIAL PLANE      MSM03400
C      DKTIME      ARRAY GIVING CHARGE EXCHANGE DECAY TIMES (SECONDS)      MSM03410
C                                     FOR AN ARRAY OF RADIAL DISTANCES, ENERGIES, SUN SPOT      MSM03420
C                                     NUMBERS, AND SPECIES      MSM03430
C                                     MSM03440
C                                     MSM03450
C                                     MSM03460
C      CHARACTER*80 CHID      MSM03470
C      INTEGER IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST, DDST,      MSM03480
C      1      EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,      MSM03490
C      2      GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,      MSM03500
C      3      STAND, PCP, DEQDT, IPATT      MSM03510
C      COMMON /ARINDX/ IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST,      MSM03520
C      1      EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,      MSM03530
C      2      GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,      MSM03540
C      3      STAND, PCP, IPATT, DDST, DEQDT      MSM03550
C      COMMON /LUNIT/ LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,      MSM03560
C      1      LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,      MSM03570
C      2      LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,      MSM03580
C      3      LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH      MSM03590
C      COMMON /MOVE/ IBEG, ISTOP, JBEG, JSTOP, IELEG, IEEND      MSM03600
C      COMMON /IRDREC/ IRECPT, IRECEB, IRDBEG      MSM03610
C      DIMENSION ISTART(3), IINC(3), IEND(3), ITIME(3),      MSM03620
C      1      ETA (LATDIM, LTDIM, IEDIM), EFLUX (LATDIM, LTDIM, IEDIM),      MSM03630
C      2      FLXSUM (LATDIM, LTDIM, 3), EAVG (LATDIM, LTDIM, 3),      MSM03640
C      3      FLUX (LATDIM, LTDIM, IEDIM)      MSM03650
C      DIMENSION VM (LATDIM, LTDIM, ITMDIM), V (LATDIM, LTDIM, ITMDIM),      MSM03660
C      1      ETABEG (LATDIM, LTDIM, IEDIM), ETABND (LTDIM, IEDIM, ITMDIM),      MSM03670
C      2      FLXBND (LTDIM, IEDIM, ITMDIM),      MSM03680
C      3      ALAM (IEDIM), COLAT (LATDIM, LTDIM), TIMTAG (ITMDIM),      MSM03690
C      4      ALOCT (LATDIM, LTDIM), BMIN (LATDIM, LTDIM), XMIN (LATDIM, LTDIM),      MSM03700
C      5      YMIN (LATDIM, LTDIM), ZMIN (LATDIM, LTDIM), BNDLOC (LTDIM, ITMDIM),      MSM03710
C      6      ALMDEL (IEDIM)      MSM03720
C      DIMENSION IFLAV (IEDIM), BNDMAX (LTDIM),      MSM03730
C      1      ALPHA (LATDIM), BETA (LATDIM), BIR (LATDIM, LTDIM),      MSM03740
C      2      ASOUTH (3, ITMDIM), BSOUTH (3, ITMDIM), DXS (3, ITMDIM), DYS (3, ITMDIM),      MSM03750
C      3      ANORTH (3, ITMDIM), BNORTH (3, ITMDIM), DXN (3, ITMDIM), DYN (3, ITMDIM),      MSM03760
C      4      VMLOSS (LATDIM, LTDIM, ITMDIM),      MSM03770
C      MSM03780
C      MSM03790
C      MSM03800
C      MSM03810
C      MSM03820
C      MSM03830
C      MSM03840
```

90/06/30
12:03:44

msm1.for

7

```

5   TETA (LATDIM) , PHI (LTDIM) , SINI (LATDIM, LTDIM)
C
C   DIMENSION VNORTH (LATDIM, LTDIM) , VSOUTH (LATDIM, LTDIM)
C
C   DIMENSION R (LATDIM, LTDIM, ITMDIM) , P (LATDIM, LTDIM, ITMDIM)
C
C   DIMENSION SCTFRC (IEDIM, LTDIM, ITMDIM) , RPP (ITMDIM)
C
C   DIMENSION AUGPAR (NAUGEL, ITMDIM)
C   LOGICAL*1 MODE (NAUGEL, ITMDIM)
C
C   DIMENSION IDGHV (2) , ID (20) , RID (20) , THRSH (IEDIM)
C   DIMENSION ERSHFT (IEDIM)
C   DIMENSION ITMNEW (3) , IRESTR (3)
C
C   DIMENSION FLXMAT (IRDIM, NRGDIM, KPPLUS, ISPDIM)
C   DIMENSION THRMAT (IRDIM, NRGDIM, KPDIM)
C   DIMENSION FLXR (IRDIM) , FLXNRG (NRGDIM) , FLXKP (KPDIM)
C
C   DIMENSION DKTIME (IRDK, INRGDK, ISOLDK, IONDK)
C
C   DATA TSHENG, TSHING /100.0, 50./
C
C   SET UP THE GRID
C
C   CALL RDGRID (LATDIM, LTDIM, ALPHA, BETA, TETA, PHI, COLAT, BIR, SINI)
C
C   ***** WRITE ALL VALUES COMPUTED IN RDGRID
C
C   WRITE (6,*) 'LATDIM = ', LATDIM
C   WRITE (6,*) 'LTDIM = ', LTDIM
C   WRITE (6,*) 'ALPHA ', ALPHA
C   WRITE (6,*) 'BETA ', BETA
C   WRITE (6,*) 'TETA ', TETA
C   WRITE (6,*) 'PHI ', PHI
C   CALL OUTP (COLAT, LATDIM, LTDIM, 1, LATDIM, 1, 3, LTDIM, 3, 0.,
1   'COLAT', 6, 132)
C   CALL OUTP (BIR, LATDIM, LTDIM, 10, LATDIM, 1, 1, LTDIM, 1, 0., 'BIR', 6, 132)
C   1   )
C   CALL OUTP (SINI, LATDIM, LTDIM, 10, LATDIM, 2, 3, LTDIM, 3, 0.,
C   1   'SINI', 6, 80)
C
C   READ TABLE OF CHARGE EXCHANGE DECAY TIMES FOR USE IN COMPUTING
C   ION CHARGE EXCHANGE LOSS
C
C   CALL READDK (IRDK, INRGDK, ISOLDK, IONDK, DKTIME)
C
C   WRITE OUT DECAY TIMES
C   DO 4 IR=1, IARDK
C     ELVAL=1.+5*FLOAT (IR)
C     WRITE (6, 865) ELVAL
865  FORMAT (1X/1X, 'L-VALUE = ', F5.2)
C     DO 6 ION=1, 2
C       IF (ION.EQ.1) WRITE (6, 866)
C       IF (ION.EQ.2) WRITE (6, 867)
866  FORMAT (1X, 'H+ DECAY TIMES: (SECONDS)')
867  FORMAT (1X, 'O+ DECAY TIMES: (SECONDS)')
C       WRITE (6, 868) ((DKTIME (IR, INRG, ISOL, ION) , INRG=1, INRGDK) ,
2       ISOL=1, ISOLDK)
868  FORMAT (13 (1X, 1PE9.3))
6   CONTINUE
```

MSM03850
MSM03860
MSM03870
MSM03880
MSM03890
MSM03900
MSM03910
MSM03920
MSM03930
MSM03940
MSM03950
MSM03960
MSM03970
MSM03980
MSM03990
MSM04000
MSM04010
MSM04020
MSM04030
MSM04040
MSM04050
MSM04060
MSM04070
MSM04080
MSM04090
MSM04100
MSM04110
MSM04120
MSM04130
MSM04140
MSM04150
MSM04160
MSM04170
MSM04180
MSM04190
MSM04200
MSM04210
MSM04220
MSM04230
MSM04240
MSM04250
MSM04260
MSM04270
MSM04280
MSM04290
MSM04300
MSM04310
MSM04320
MSM04330
MSM04340
MSM04350
MSM04360
MSM04370
MSM04380
MSM04390
MSM04400
MSM04410
MSM04420
MSM04430
MSM04440
MSM04450
MSM04460
MSM04470
MSM04480


```

4      CONTINUE
C
C
C      SET UP TIME INDEPENDENT ENERGY CHANNELS TO MODEL
C
C      CALL SETALM(ALAM,IEMAX,IEDIM,IFLAV,ALMDEL)
C      IEEND=IEMAX
C
C      WRITE OUT ENERGY CHANNEL INFORMATION
C      DO 999 IE=1,IEMAX
C          WRITE(6,*) 'IFLAV=',IFLAV(IE),' ALAM=',ALAM(IE),
C              2      ' ALMDEL=',ALMDEL(IE)
C          THRS(IE)=0.0
999      CONTINUE
C
C      INITIALIZE MATRIX OF KP DEPENDENT DEFAULT FLUXES
C
C      CALL FLXNIT(IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,
C          2      FLXMAT,THRMAT)
C
C      SPECIAL PRINTOUT FOR FLXMAT AND THRMAT
C      DO 300 IR=1,IRDIM
C          IF(IR.EQ.1) RDIST=3.
C          IF(IR.EQ.2) RDIST=4.
C          IF(IR.EQ.3) RDIST=6.6
C          IF(IR.EQ.4) RDIST=13.
C
C          DO 310 IKP=1,KPPLUS
C              MYKP=IKP-1
C              WRITE(6,850) RDIST,MYKP
850      FORMAT(1X,'R=',F6.2,2X,'KP=',I2,/)
C              WRITE(6,*) 'ELECTRONS'
C              WRITE(6,851)
851      FORMAT(1X,'LOG ENERGY',5X,'LOG FLUX',5X,'LOG THRESHOLD')
C
C          DO 320 INRG=1,NRGDIM
C              WRITE(6,852) FLXNRG(INRG),FLXMAT(IR,INRG,IKP,1),
C              2      THRMAT(IR,INRG,IKP)
852      FORMAT(F10.4,5X,F8.4,5X,F13.4)
C
C          320      CONTINUE
C
C              WRITE(6,853)
853      FORMAT(1X/)
C
C              WRITE(6,*) 'H+ IONS'
C              DO 321 INRG=1,NRGDIM
C                  WRITE(6,859) FLXNRG(INRG),FLXMAT(IR,INRG,IKP,2)
859      FORMAT(F10.4,5X,F8.4)
C
C          321      CONTINUE
C
C              WRITE(6,853)
C
C              WRITE(6,*) 'O+ IONS'
C              DO 322 INRG=1,NRGDIM
C                  WRITE(6,859) FLXNRG(INRG),FLXMAT(IR,INRG,IKP,3)
C
C          322      CONTINUE
C
C              WRITE(6,853)
C
C          310      CONTINUE

```

```

MSM04490
MSM04500
MSM04510
MSM04520
MSM04530
MSM04540
MSM04550
MSM04560
MSM04570
MSM04580
MSM04590
MSM04600
MSM04610
MSM04620
MSM04630
MSM04640
MSM04650
MSM04660
MSM04670
MSM04680
MSM04690
MSM04700
MSM04710
MSM04720
MSM04730
MSM04740
MSM04750
MSM04760
MSM04770
MSM04780
MSM04790
MSM04800
MSM04810
MSM04820
MSM04830
MSM04840
MSM04850
MSM04860
MSM04870
MSM04880
MSM04890
MSM04900
MSM04910
MSM04920
MSM04930
MSM04940
MSM04950
MSM04960
MSM04970
MSM04980
MSM04990
MSM05000
MSM05010
MSM05020
MSM05030
MSM05040
MSM05050
MSM05060
MSM05070
MSM05080
MSM05090
MSM05100
MSM05110
MSM05120

```

90/06/30
12:03:44

msml.for

9

300 CONTINUE

C
C

```
WRITE(6,*) 'ELECTRONS'
WRITE(6,*) 'FLXMAT ARRAYS (IR,NRG,IKP); IR=1 IS 3RE, IR=2 IS 4RE'
WRITE(6,*) 'IR=3 IS 6.6 RE, IR=4 IS 13 RE; NRG CORRESPONDS TO'
WRITE(6,*) 'LOG OF ENERGY IN EV RUNNING FROM 10 EV TO 10**8 EV'
WRITE(6,*) 'ENERGY= 10.**((3+NRG)/4); KP=IKP-1'
WRITE(6,*) 'KP=0'
CALL OUTP (FLXMAT(1,1,1,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT1', 6, 132)
WRITE(6,*) 'KP=1'
CALL OUTP (FLXMAT(1,1,2,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT1', 6, 132)
WRITE(6,*) 'KP=2'
CALL OUTP (FLXMAT(1,1,3,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT1', 6, 132)
WRITE(6,*) 'KP=3'
CALL OUTP (FLXMAT(1,1,4,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT1', 6, 132)
WRITE(6,*) 'KP=4'
CALL OUTP (FLXMAT(1,1,5,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT1', 6, 132)
WRITE(6,*) 'KP=5'
CALL OUTP (FLXMAT(1,1,6,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT1', 6, 132)
WRITE(6,*) 'KP=6'
CALL OUTP (FLXMAT(1,1,7,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT1', 6, 132)
WRITE(6,*) 'MINIMA'
CALL OUTP (FLXMAT(1,1,8,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT1', 6, 132)
WRITE(6,*) 'MAXIMA'
CALL OUTP (FLXMAT(1,1,9,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT1', 6, 132)
WRITE(6,*) 'H+ IONS'
WRITE(6,*) 'KP=0'
CALL OUTP (FLXMAT(1,1,1,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT2', 6, 132)
WRITE(6,*) 'KP=1'
CALL OUTP (FLXMAT(1,1,2,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT2', 6, 132)
WRITE(6,*) 'KP=2'
CALL OUTP (FLXMAT(1,1,3,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT2', 6, 132)
WRITE(6,*) 'KP=3'
CALL OUTP (FLXMAT(1,1,4,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT2', 6, 132)
WRITE(6,*) 'KP=4'
CALL OUTP (FLXMAT(1,1,5,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT2', 6, 132)
WRITE(6,*) 'KP=5'
CALL OUTP (FLXMAT(1,1,6,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT2', 6, 132)
WRITE(6,*) 'KP=6'
CALL OUTP (FLXMAT(1,1,7,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT2', 6, 132)
WRITE(6,*) 'MINIMA'
CALL OUTP (FLXMAT(1,1,8,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT2', 6, 132)
WRITE(6,*) 'MAXIMA'
CALL OUTP (FLXMAT(1,1,9,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2 'FLXMAT2', 6, 132)
WRITE(6,*) 'O+ IONS'
```

MSM05130
MSM05140
MSM05150
MSM05160
MSM05170
MSM05180
MSM05190
MSM05200
MSM05210
MSM05220
MSM05230
MSM05240
MSM05250
MSM05260
MSM05270
MSM05280
MSM05290
MSM05300
MSM05310
MSM05320
MSM05330
MSM05340
MSM05350
MSM05360
MSM05370
MSM05380
MSM05390
MSM05400
MSM05410
MSM05420
MSM05430
MSM05440
MSM05450
MSM05460
MSM05470
MSM05480
MSM05490
MSM05500
MSM05510
MSM05520
MSM05530
MSM05540
MSM05550
MSM05560
MSM05570
MSM05580
MSM05590
MSM05600
MSM05610
MSM05620
MSM05630
MSM05640
MSM05650
MSM05660
MSM05670
MSM05680
MSM05690
MSM05700
MSM05710
MSM05720
MSM05730
MSM05740
MSM05750
MSM05760

90/06/30
12:03:44

msm1.for

10

```
WRITE(6,*) 'KP=0'
CALL OUTP (FLXMAT(1,1,1,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'FLXMAT3', 6, 132)
WRITE(6,*) 'KP=1'
CALL OUTP (FLXMAT(1,1,2,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'FLXMAT3', 6, 132)
WRITE(6,*) 'KP=2'
CALL OUTP (FLXMAT(1,1,3,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'FLXMAT3', 6, 132)
WRITE(6,*) 'KP=3'
CALL OUTP (FLXMAT(1,1,4,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'FLXMAT3', 6, 132)
WRITE(6,*) 'KP=4'
CALL OUTP (FLXMAT(1,1,5,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'FLXMAT3', 6, 132)
WRITE(6,*) 'KP=5'
CALL OUTP (FLXMAT(1,1,6,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'FLXMAT3', 6, 132)
WRITE(6,*) 'KP=6'
CALL OUTP (FLXMAT(1,1,7,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'FLXMAT3', 6, 132)
WRITE(6,*) 'MINIMA'
CALL OUTP (FLXMAT(1,1,8,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'FLXMAT3', 6, 132)
WRITE(6,*) 'MAXIMA'
CALL OUTP (FLXMAT(1,1,9,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'FLXMAT3', 6, 132)
CALL OUTP (THRMAT(1,1,1), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'THRMAT1', 6, 80)
CALL OUTP (THRMAT(1,1,2), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'THRMAT2', 6, 80)
CALL OUTP (THRMAT(1,1,3), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'THRMAT3', 6, 80)
CALL OUTP (THRMAT(1,1,4), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'THRMAT4', 6, 80)
CALL OUTP (THRMAT(1,1,5), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'THRMAT5', 6, 80)
CALL OUTP (THRMAT(1,1,6), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'THRMAT6', 6, 80)
CALL OUTP (THRMAT(1,1,7), IRDIM, NRGDIM, 1, IRDIM, 1, 1, NRGDIM, 1, 0.,
2      'THRMAT7', 6, 80)

C
C GET CURRENT TIME
C
C CALL DTIME(ISTEB)
C
C
C IRECST=MAX(1, IRECEB)
C
C CALL TO EBCON COMPUTES ALL THE ELECTRIC AND MAGNETIC FIELD MODELS
C   NEEDED FOR THE RUN
C
C CALL EBCON(ISTART, IINC, IEND, COLAT, ALOCT, LATDIM, LTDIM, ITMDIM,
1      IEDIM, NAUGEL, CHID, ID, RID, VM, BMIN, XMIN, YMIN, ZMIN,
2      V, VNORTH, VSOUTH, ASOUTH, BSOUTH, DXS, DYS,
3      ANORTH, BNORTH, DXN, DYN, BNDLOC,
4      TIMTAG, ITMMAX, R, P, BIR, SINI, VMLOSS, SCTFRC, AUGPAR, MODE,
5      RPP, IEMAX, TETA)

C
C GET CURRENT TIME AND FIND TIME SPENT IN EBCON
C
C CALL DTIME(IENDEB)
C WRITE (LUERR,*) 'THE TIME SPENT IN EBCON IS ',
C 1      ABS(IENDEB-ISTEB)/100., ' SECONDS.'
```

MSM05770
MSM05780
MSM05790
MSM05800
MSM05810
MSM05820
MSM05830
MSM05840
MSM05850
MSM05860
MSM05870
MSM05880
MSM05890
MSM05900
MSM05910
MSM05920
MSM05930
MSM05940
MSM05950
MSM05960
MSM05970
MSM05980
MSM05990
MSM06000
MSM06010
MSM06020
MSM06030
MSM06040
MSM06050
MSM06060
MSM06070
MSM06080
MSM06090
MSM06100
MSM06110
MSM06120
MSM06130
MSM06140
MSM06150
MSM06160
MSM06170
MSM06180
MSM06190
MSM06200
MSM06210
MSM06220
MSM06230
MSM06240
MSM06250
MSM06260
MSM06270
MSM06280
MSM06290
MSM06300
MSM06310
MSM06320
MSM06330
MSM06340
MSM06350
MSM06360
MSM06370
MSM06380
MSM06390
MSM06400

90/06/30
12:03:44

msml.for

11

```
C
C      COMPUTE BNDMAX FROM BNDLOC
C
      DO 8 LT=1,LTDIM
        BNDMAX(LT)=0.
        DO 9 ITM=1,ITMMAX
          IF (BNDLOC(LT,ITM).GT.BNDMAX(LT))
            BNDMAX(LT)=BNDLOC(LT,ITM)
1          CONTINUE
9        CONTINUE
8      CONTINUE
C
C
C
C
C
C
C
C      FIND THE TIME DEPENDENT PLASMA BOUNDARY CONDITION FOR PPTM
C
      CALL BNDSET(LATDIM,LTDIM,IEDIM,ITMDIM,ITMMAX,
1      IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,
2      NAUGEL,TIMTAG,BNDLOC,IFLAV,AUGPAR,R,P,VM,ETA,ETABND,
3      FLXBND,ALAM,ALMDEL,IEMAX)
C
C
      CALL OUTP(FLXBND(1,1,1),LTDIM,IEDIM,3,LTDIM,1,1,IEMAX,1,
2      0.,'FLXBND(J,IE,1)',6,132)
C
C
      IF (IRECPT.EQ.0) THEN
        AZERO=.2
        BZERO=0.
        CZERO=0.
      ELSE
C
C      READ START FLUX FROM DATA SET
      IRECMX=ABS(IRECPT)
      IRDFLX=(IRECMX-1)*IEMAX+1
      CALL RDHDR(LUFLX,'AFLX ',IRDFLX,ID,RID,CHID,
2      LATDIM,LTDIM,ITMDIM)
C
C      CHECK IF TIME IS CORRECT
C
      ISEC=ID(3)*3600+ID(4)*60+ID(5)
      IF (ID(1).NE.ISTART(1).OR.
1      ID(2).NE.ISTART(2).OR.
2      ISEC.NE.ISTART(3)) THEN
C
        WRITE(6,*) 'START TIME DOES NOT MATCH ETABEG START TIME'
        WRITE(6,*) 'STOPPING PROGRAM IN PPTCON'
        WRITE(6,*) 'ISTART=',ISTART
        WRITE(6,*) 'ISEC=',ISEC,'IRECMX=',IRECMX
C
        STOP
        END IF
C
      CALL READ3D(LUFLX,'AFLX ',IRECMX,LATDIM,LTDIM,IEMAX,ITMDIM,
1      ID,RID,CHID,FLUX)
C
C      CONVERT FROM PARTICLE FLUX TO INVARIANT FLUX ETA
C
      CALL FLX2ET(LATDIM,LTDIM,IEDIM,IEMAX,ITMDIM,1,IFLAV,ALAM,
2      ALMDEL,BNDLOC,VM,FLUX,ETA)
```

MSM06410
MSM06420
MSM06430
MSM06440
MSM06450
MSM06460
MSM06470
MSM06480
MSM06490
MSM06500
MSM06510
MSM06520
MSM06530
MSM06540
MSM06550
MSM06560
MSM06570
MSM06580
MSM06590
MSM06600
MSM06610
MSM06620
MSM06630
MSM06640
MSM06650
MSM06660
MSM06670
MSM06680
MSM06690
MSM06700
MSM06710
MSM06720
MSM06730
MSM06740
MSM06750
MSM06760
MSM06770
MSM06780
MSM06790
MSM06800
MSM06810
MSM06820
MSM06830
MSM06840
MSM06850
MSM06860
MSM06870
MSM06880
MSM06890
MSM06900
MSM06910
MSM06920
MSM06930
MSM06940
MSM06950
MSM06960
MSM06970
MSM06980
MSM06990
MSM07000
MSM07010
MSM07020
MSM07030
MSM07040

90/06/30
12:03:44

msml.for

12

```
C
C
C
      AZERO=0.
      BZERO=.2
      CZERO=0.
C
      END IF
C
      IRECPT=MAX(1, IRECPT)
      LL=1
C
C
C INITIALIZE PLASMA DISTRIBUTION
C
      CALL INITAL(LL, LATDIM, LTDIM, IEDIM, ITMDIM, IRDIM,
2          NRGDIM, KPDIM, KPPLUS, ISPDIM, FLXR, FLXNRG, FLXKP,
3          FLXMAT, IFLAV, AUGPAR(KP, LL), R, P, VM, ETA, ETABEG, ALAM, ALMDEL,
4          BNDLOC, AZERO, BZERO, CZERO, IEMAX)
C
C CONVERT ETABEG TO FLUX AND WRITE OUT TO DISK FILE
      CALL ET2FLX(LATDIM, LTDIM, IEDIM, IEMAX, ITMDIM, 1, IFLAV, ALAM,
2          ALMDEL, BNDLOC, VM, FLUX, ETABEG)
C
      ID(1)=ISTART(1)
      ID(2)=ISTART(2)
      ID(3)=ISTART(3)/3600
      ID(4)=MOD(ISTART(3), 3600)/60
      ID(5)=MOD(ISTART(3), 60)
      ID(11)=1
      ID(8)=LATDIM
      ID(9)=LTDIM
      ID(10)=IEMAX
C
      CALL WRT3D(LUFLX, IRECPT, ID, RID, CHID, FLUX, LATDIM, LTDIM, IEMAX,
2          ITMDIM)
C
C
      DO 400 KK=1, IEMAX
        WRITE(6, *) 'KK= ', KK
        CALL OUTP(ETABEG(1, 1, KK), LATDIM, LTDIM, 20, 45, 1, 3, LTDIM-1,
2          1, 0., 'ETABEG(KK)', 6, 132)
        WRITE(6, *) 'KK= ', KK, 'BEGINNING FLUX'
        CALL OUTP(FLUX(1, 1, KK), LATDIM, LTDIM, 20, 45, 1, 3, LTDIM-1,
2          1, 0., 'FLUX(KK)', 6, 132)
400 CONTINUE
C
C
      ITIME(1) = ISTART(1)
      ITIME(2) = ISTART(2)
      ITIME(3) = ISTART(3)
C
C
C
C
C FOLLOWING IS THE MAIN TIME LOOP IN THE PROGRAM
C LOOP OVER ALL TIMES IN TIMTAG (L=1 CORRESPONDS TO INITIAL TIME)
      DO 10 L=2, ITMMAX
        IRECPT=IRECPT+1
        TSTART=TIMTAG(L)
        TSTOP=TIMTAG(L-1)
        PSTEP=TSTART-TSTOP
        WRITE(6, *) ' L= ', L, ' IRECPT= ', IRECPT, ' TSTART= ', TSTART
```

MSM07050
MSM07060
MSM07070
MSM07080
MSM07090
MSM07100
MSM07110
MSM07120
MSM07130
MSM07140
MSM07150
MSM07160
MSM07170
MSM07180
MSM07190
MSM07200
MSM07210
MSM07220
MSM07230
MSM07240
MSM07250
MSM07260
MSM07270
MSM07280
MSM07290
MSM07300
MSM07310
MSM07320
MSM07330
MSM07340
MSM07350
MSM07360
MSM07370
MSM07380
MSM07390
MSM07400
MSM07410
MSM07420
MSM07430
MSM07440
MSM07450
MSM07460
MSM07470
MSM07480
MSM07490
MSM07500
MSM07510
MSM07520
MSM07530
MSM07540
MSM07550
MSM07560
MSM07570
MSM07580
MSM07590
MSM07600
MSM07610
MSM07620
MSM07630
MSM07640
MSM07650
MSM07660
MSM07670
MSM07680

90/06/30
12:03:44

13

msml.for

```
C      WRITE(6,*) ' TSTOP= ',TSTOP,' PSTEP= ',PSTEP,' ITMMAX=',ITMMAX      MSM07690
C      CALL OUTP(ETABND(1,1,L),LTDIM,IEDIM,3,LTDIM,1,1,IEMAX,1,      MSM07700
2      0.,'ETABND(J,IE,L)',6,132)      MSM07710
C      MSM07720
C      MSM07730
C      MSM07740
C      DTNOM=-600.      MSM07750
C      IF (ABS(DTNOM).GT.PSTEP) DTNOM=-PSTEP      MSM07760
C      MSM07770
C      LL=L      MSM07780
C      MSM07790
C      MSM07800
C      MSM07810
C      MSM07820
C      IMODE=0      MSM07830
C      MSM07840
C      IF (MODE(KP,L).AND.IMODE.EQ.0) THEN      MSM07850
C      PERFORM FULL PARTICLE TRACES      MSM07860
C      MSM07870
C      GET THE CURRENT TIME      MSM07880
C      CALL DTIME(ISTPPT)      MSM07890
C      MSM07900
C      CALL PPTM(TSTART,TSTOP,DTNOM,SCTFRC,IFLAV,BNDMAX,ALPHA,BETA,      MSM07910
1      BIR,ITMMAX,LATDIM,LTDIM,IEDIM,ITMDIM,IRDIM,NRGDIM,KPDIM,      MSM07920
2      KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,THRMAT,TIMTAG,ALAM,      MSM07930
2      ALMDEL,VM,R,P,VMLOSS,V,ETABEG,ETABND,BNDLOC,ETA,EFLUX,      MSM07940
3      FLXSUM,EAVG,LL,RPP,AUGPAR(KP,LL),IEMAX,THRSH,TSHENG,      MSM07950
4      TSHING,IRDK,INRGDK,ISOLDK,IONDK,DKTIME)      MSM07960
C      MSM07970
C      GET THE CURRENT TIME AND FIND THE TIME SPENT IN PPTM      MSM07980
C      MSM07990
C      CALL DTIME(IENPPT)      MSM08000
C      WRITE(LUERR,*)' THE TIME SPENT IN PPTM IS ',      MSM08010
1      ABS(IENPPT-ISTPPT)/100.,' SECONDS.'      MSM08020
C      MSM08030
C      CALL PWRCAL TO ESTIMATE ELECTRON PRECIPITATION POLEWARD OF MAIN      MSM08040
C      MSM MODELING REGION      MSM08050
C      MSM08060
C      CALL PWRCAL(LATDIM,LTDIM,3,ITMDIM,LL,AUGPAR(KP,LL),FLXSUM,      MSM08070
2      EAVG,COLAT,ALOCT,ALPHA,BETA,BNDLOC,ANORTH,      MSM08080
3      BNORTH,DXN,DYN)      MSM08090
C      MSM08100
C      WRITE(18,*)' TIME = ',TIMTAG(LL)      MSM08110
C      WRITE(6,*)' TIME = ',TIMTAG(LL)      MSM08120
C      MSM08130
C      ELSE      MSM08140
C      MSM08150
C      USE EMPIRICAL FLXMAT DENSITIES      MSM08160
C      MSM08170
C      CALL DTIME(ISTFLX)      MSM08180
C      MSM08190
C      CALL FLXDFL(LL,AUGPAR(KP,LL),R,P,ALAM,VM,IRDIM,NRGDIM,      MSM08200
1      KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP,EFLUX,ETA,      MSM08210
2      ALMDEL,LATDIM,LTDIM,IEDIM,ITMDIM,BNDLOC,IEMAX,IFLAV,      MSM08220
3      1,IEMAX)      MSM08230
C      MSM08240
C      DO 998 IDO=1,IEMAX      MSM08250
C      IF (IFLAV(IDO).EQ.1) THEN      MSM08260
C      THRSH(IDO)=TSHENG      MSM08270
C      ELSE      MSM08280
C      THRSH(IDO)=TSHING      MSM08290
C      END IF      MSM08300
998      CONTINUE      MSM08310
C      MSM08320
```

90/06/30
12:03:44

msml.for

14

```
C
C      CALL DTIME(INDFLX)
C      WRITE(LUERR,*) ' THE TIME SPEND IN FLXDFL IS ',
C      1      ABS(INDFLX-ISTFLX)/100., ' SECONDS.'
C      CALL DTIME(ISTAUR)
C      ENDIF
C
C
C      CALL TO AURL2S PRODUCES HARDY ION PRECIPITATION MODEL
C      CALL AURL2S(AUGPAR(KP,L),COLAT,ALOCT,LATDIM,LTDIM,
C      1      FLXSUM(1,1,2),EAVG(1,1,2))
C      CALL OUTP(FLXSUM(1,1,2),LATDIM,LTDIM,15,LATDIM,1,3,
C      1      LTDIM,1,0., 'IFLXSUM',6,132)
C
C      CALL DTIME(IENAUR)
C      WRITE(LUERR,*) ' THE TIME SPENT IN AURL IS ',
C      1      ABS(IENAUR-ISTAUR)/100., ' SECONDS.'
C
C      WRITE(6,*) 'T=',TSTART
C
C      CONVERT TSTOP TO 3 VALUED TIME AND PUT IT INTO ITIME
C      GET ALOCT AT TSTOP
C
C      ITIME(3) = ITIME(3) + INT(PSTEP)
C      11      CONTINUE
C      IF (ITIME(3).GT.86400) THEN
C      ITIME(2) = ITIME(2) + 1
C      ITIME(3) = ITIME(3) - 86400
C      IF (MOD(ITIME(1),4).NE.0.AND.ITIME(2).GT.365) THEN
C      ITIME(2) = ITIME(2) - 365
C      ITIME(1) = ITIME(1) + 1
C      ELSEIF (MOD(ITIME(1),4).EQ.0.AND.ITIME(2).GT.366) THEN
C      ITIME(2) = ITIME(2) - 366
C      ITIME(1) = ITIME(1) + 1
C      END IF
C      GO TO 11
C      END IF
C
C      ----> WRITE FLUX, EFLUX, FLXSUM, EAVG
C      ----> TO ARCHIVAL FILES
C
C      DO 2 IND=1,20
C      ID(IND)=0
C      RID(IND)=0.
C      2      CONTINUE
C      WRITE(6,*) 'L=',L,'ITIME=',ITIME(3)
C      ID(1)=ITIME(1)
C      ID(2)=ITIME(2)
C      ID(3)=ITIME(3)/3600
C      ID(4)=MOD(ITIME(3),3600)/60
C      ID(5)=MOD(ITIME(3),60)
C      ID(11)=L
C
C      ----> WRITE FLXBND TO ARCHIVE FILE
C
C      ID(8)=LTDIM
C      ID(9)=IEDIM
C      ID(10)=1
C
C      CALL WRT3D(LUFLXB,IRECPT,ID,RID,CHID,FLXBND(1,1,L),LTDIM,IEDIM,
C      2      1,ITMDIM)
```

MSM08330
MSM08340
MSM08350
MSM08360
MSM08370
MSM08380
MSM08390
MSM08400
MSM08410
MSM08420
MSM08430
MSM08440
MSM08450
MSM08460
MSM08470
MSM08480
MSM08490
MSM08500
MSM08510
MSM08520
MSM08530
MSM08540
MSM08550
MSM08560
MSM08570
MSM08580
MSM08590
MSM08600
MSM08610
MSM08620
MSM08630
MSM08640
MSM08650
MSM08660
MSM08670
MSM08680
MSM08690
MSM08700
MSM08710
MSM08720
MSM08730
MSM08740
MSM08750
MSM08760
MSM08770
MSM08780
MSM08790
MSM08800
MSM08810
MSM08820
MSM08830
MSM08840
MSM08850
MSM08860
MSM08870
MSM08880
MSM08890
MSM08900
MSM08910
MSM08920
MSM08930
MSM08940
MSM08950
MSM08960

90/06/30
12:03:44

msm1.for

15

```

      ID(8)=LATDIM
      ID(9)=LTDIM
      ID(10)=1
C
C
C
C
C      ----> WRITE EAVG AND FLXSUM TO ARCHIVE FILES
C
C      CALL WRT3D(LUEAVG, IRECPT, ID, RID, CHID, EAVG, LATDIM, LTDIM, 1, ITMDIM)
C      CALL WRT3D(LUFLSM, IRECPT, ID, RID, CHID, FLXSUM, LATDIM, LTDIM, 1, ITMDIM)
C      CALL WRT3D(LUPIFX, IRECPT, ID, RID, CHID, FLXSUM(1, 1, 2), LATDIM, LTDIM,
2      1, ITMDIM)
C      CALL WRT3D(LUPIEN, IRECPT, ID, RID, CHID, EAVG(1, 1, 2), LATDIM, LTDIM,
2      1, ITMDIM)
C
C
C      ----> WRITE FLUX AND EFLUX TO ARCHIVE FILES
C
C
C      ID(8)=LATDIM
C      ID(9)=LTDIM
C      ID(10)=IEMAX
C
C
C      CONVERT ETA TO FLUX FOR OUTPUT
C
C      CALL ET2FLX(LATDIM, LTDIM, IEDIM, IEMAX, ITMDIM, L, IFLAV, ALAM,
2      ALMDEL, BNDLOC, VM, FLUX, ETA)
C
C
C      CALL WRT3D(LUFLX, IRECPT, ID, RID, CHID, FLUX, LATDIM, LTDIM, IEMAX,
1      ITMDIM)
C      CALL WRT3D(LUEFLX, IRECPT, ID, RID, CHID, EFLUX, LATDIM, LTDIM, IEMAX,
C      1      ITMDIM)
C
C      IUT=100*ID(3)+ID(4)
C      DO 111 IE=1, IEMAX
C      WRITE(6,*) 'IE=', IE, ' ALAM(IE)=', ALAM(IE), ' SPECIES=', IFLAV(IE),
1      'L=', L, ' DAY=', ID(2), ' ITIME=', ITIME(3), 'UT=', IUT
C      CALL OUTP(ETA(1, 1, IE), LATDIM, LTDIM, 15, 45, 1, 3, LTDIM, 1, 0.,
C      1      'ETA', 6, 132)
C      CALL OUTP(FLUX(1, 1, IE), LATDIM, LTDIM, 15, 45, 1, 3, LTDIM, 1, 0.,
1      'FLUX', 6, 132)
111 CONTINUE
C
C      CALL OUTP(ALOCT, LATDIM, LTDIM, 10, 11, 1, 3, LTDIM, 1, 0.,
1      'ALOCT', 6, 132)
C      WRITE(6,*) 'L=', L, ' DAY=', ID(2), 'UT=', IUT, ' ITIME=', ITIME(3)
C      CALL OUTP(FLXSUM, LATDIM, LTDIM, 15, 55, 1, 3, LTDIM, 1, 0.,
1      'FLXSUM', 6, 132)
C      WRITE(6,*) 'L=', L, ' DAY=', ID(2), 'UT=', IUT, ' ITIME=', ITIME(3)
C      CALL OUTP(EAVG, LATDIM, LTDIM, 15, 55, 1, 3, LTDIM, 3, 0.,
1      'EAVG', 6, 132)
C
C
C      COPY CURRENT VALUES IN ETA TO ETABEG
C
C      DO 20, I=1, LATDIM
C      DO 30, J=1, LTDIM
C      DO 40, K=1, IEMAX
C      ETABEG(I, J, K) = ETA(I, J, K)
40      CONTINUE
30      CONTINUE
20      CONTINUE
C
10 CONTINUE
C      CALL OPTERR(ETA, VM, LATDIM, LTDIM, IEDIM, IEMAX, ALAM, ALMDEL, THRSH,
1      ITMDIM, IFLAV, ISTART, IEND, R, P, TIMTAG, ITMMAX, ERSHT)
C
```

MSMJ8970
MSM08980
MSM08990
MSM09000
MSM09010
MSM09020
MSM09030
MSM09040
MSM09050
MSM09060
MSM09070
MSM09080
MSM09090
MSM09100
MSM09110
MSM09120
MSM09130
MSM09140
MSM09150
MSM09160
MSM09170
MSM09180
MSM09190
MSM09200
MSM09210
MSM09220
MSM09230
MSM09240
MSM09250
MSM09260
MSM09270
MSM09280
MSM09290
MSM09300
MSM09310
MSM09320
MSM09330
MSM09340
MSM09350
MSM09360
MSM09370
MSM09380
MSM09390
MSM09400
MSM09410
MSM09420
MSM09430
MSM09440
MSM09450
MSM09460
MSM09470
MSM09480
MSM09490
MSM09500
MSM09510
MSM09520
MSM09530
MSM09540
MSM09550
MSM09560
MSM09570
MSM09580
MSM09590
MSM09600

90/06/30
12:03:44

msm1.for

16

RETURN
END

MSM09610
MSM09620
MSM09630
MSM09640
MSM09650
MSM09660
MSM09670
MSM09680
MSM09690
MSM09700
MSM09710
MSM09720
MSM09730
MSM09740
MSM09750
MSM09760
MSM09770
MSM09780
MSM09790
MSM09800
MSM09810
MSM09820
MSM09830
MSM09840
MSM09850
MSM09860
MSM09870
MSM09880
MSM09890
MSM09900
MSM09910
MSM09920
MSM09930
MSM09940
MSM09950
MSM09960
MSM09970
MSM09980
MSM09990
MSM10000
MSM10010
MSM10020
MSM10030
MSM10040
MSM10050
MSM10060
MSM10070
MSM10080
MSM10090
MSM10100
MSM10110
MSM10120
MSM10130
MSM10140
MSM10150
MSM10160
MSM10170
MSM10180
MSM10190
MSM10200
MSM10210
MSM10220
MSM10230
MSM10240

```
C
C ***** E B C O N *****
C
  SUBROUTINE EBCON (ISTART, IINC, IEND, COLAT, ALOCT, LATDIM, LTDIM, ITMDIM,
1    IEDIM, NAUGEL, CHID, ID, RID, VM, BMIN, XMIN, YMIN, ZMIN, V, VNORTH,
2    VSOUTH, ASOUTH, BSOUTH, DXS, DYS, ANORTH, BNORTH, DXN, DYN, BNDLOC,
3    TIMTAG, ITMMAX, R, P, BIR, SINI, VMLOSS, SCTFRC, AUGPAR, MODE, RPP,
4    IEMAX, TETA)

C
C  VERSION 1.1                      DATE: FEBRUARY 5, 1988
C                                  09.12.89
C
C  PROGRAMMER: BRYAN A. BALES/R.W. SPIRO
C
C  PURPOSE:  EBCON CALLS ROUTINES TO CALCULATE THE ELECTRIC AND THE
C            MAGNETIC FIELD AT OUR GRID POINTS.
C
C            CHARACTER*80 CHID
C
C            INTEGER IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST, DDST,
1    EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2    GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3    STAND, PCP, DEQDT, IPATT

C
C            COMMON /ARINDX/ IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST,
1    EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2    GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3    STAND, PCP, IPATT, DDST, DEQDT

C
C            COMMON /LUNIT/ LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,
1    LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,
2    LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,
3    LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH

C
C            COMMON /IRDREC/ IRECPT, IRECEB, IRDBEG

C
C            DIMENSION AUGPAR (NAUGEL, ITMDIM)
C            DIMENSION ISTART (3), IINC (3), IEND (3), ITIME (3),
1    ITMNEW (3), VM (LATDIM, LTDIM, ITMDIM), V (LATDIM, LTDIM, ITMDIM),
2    BMIN (LATDIM, LTDIM), XMIN (LATDIM, LTDIM), TIMTAG (ITMDIM),
3    YMIN (LATDIM, LTDIM), ZMIN (LATDIM, LTDIM),
4    COLAT (LATDIM, LTDIM), ALOCT (LATDIM, LTDIM)

C
C            DIMENSION VNORTH (LATDIM, LTDIM), VSOUTH (LATDIM, LTDIM)
C            DIMENSION TETA (LATDIM)

C
C            DIMENSION BNDLOC (LTDIM, ITMDIM),
2    ASOUTH (3, ITMDIM), BSOUTH (3, ITMDIM), DXS (3, ITMDIM), DYS (3, ITMDIM),
3    ANORTH (3, ITMDIM), BNORTH (3, ITMDIM), DXN (3, ITMDIM), DYN (3, ITMDIM),
4    BIR (LATDIM, LTDIM), SINI (LATDIM, LTDIM),
5    VMLOSS (LATDIM, LTDIM, ITMDIM), SCTFRC (IEDIM, LTDIM, ITMDIM),
6    RPP (ITMDIM)

C
C            DIMENSION R (LATDIM, LTDIM, ITMDIM), P (LATDIM, LTDIM, ITMDIM)

C
C            LOGICAL*1 MODE (NAUGEL, ITMDIM)

C
C            DIMENSION ID (20), RID (20)

C
C            ITIME (IYEAR) = ISTART (IYEAR)
```

90/06/30
12:03:44

msml.for

17

```

      ITIME(IDAY) = ISTART(IDAY)
      ITIME(ISECND) = ISTART(ISECND)
C
C
      PI=ATAN2(0.,-1.)
C
C  INITIALIZE ID ARRAYS
      DO 4 IC=1,20
        ID(IC)=0
        RID(IC)=0.
      4  CONTINUE
C
C  FECON RETURNS TIME-DEPENDENT INPUT PARAMETERS
C
C      CALL DTIME(ISTFE)
C
C      CALL FECON(ISTART,IINC,IEND,ITMDIM,NAUGEL,MODE,AUGPAR,
      2      ITMMAX,TIMTAG)
C
      WRITE (6,*) 'ITMMAX = ', ITMMAX
      DO 101, I=1, ITMMAX
        WRITE (6,800) I,TIMTAG(I)
      101  CONTINUE
      800  FORMAT (' TIMTAG(',I2.2,') = ',F15.1)
C
C  GET THE CURRENT TIME AND FIND THE TIME SPENT IN FECON
C
C      CALL DTIME(IENDFE)
C      WRITE (LUERR,*) 'THE TIME SPENT IN FECON IS ',
      1      ABS(IENDFE-ISTFE)/100.,' SECONDS.'
C
C
C  BEGINNING OF EB TIME LOOP
      IRECEB=MAX(1,IRECEB)
      IRECEB=IRECEB-1
C
      DO 10 IT=1,ITMMAX
        L = IT
        IRECEB=IRECEB+1
C
C      SET ALOCT TO CORRECT VALUE
C
C      CALL MLTSET(ITIME(ISECND),LATDIM,LTDIM,ALOCT)
C
C  CALCULATE SCTFRC ARRAY = FRACTION OF STRONG PITCH ANGLE SCATTERING
C
C      FKP=AUGPAR(KP,L)
C      CALL SETSCT(L,ITIME(ISECND),IEDIM,LTDIM,ITMDIM,LATDIM,
      2      FKP,ALAM,ALOCT,SCTFRC,IEMAX)
C
C
C      ***** PRINT OUT ALOCT
C
C      WRITE (6,*) 'MY UT =', (ITIME(ISECND)/3600)*100+
      1      MOD(ITIME(ISECND),3600)/60
C      CALL OUTP(ALOCT,LATDIM,LTDIM,10,11,1,3,LTDIM,1,0.,
      1      'ALOCT',6,132)
C
C  GET THE CURRENT TIME
C
C
C      ***** WRITE AUGPAR
C
```

MSM10250
MSM10260
MSM10270
MSM10280
MSM10290
MSM10300
MSM10310
MSM10320
MSM10330
MSM10340
MSM10350
MSM10360
MSM10370
MSM10380
MSM10390
MSM10400
MSM10410
MSM10420
MSM10430
MSM10440
MSM10450
MSM10460
MSM10470
MSM10480
MSM10490
MSM10500
MSM10510
MSM10520
MSM10530
MSM10540
MSM10550
MSM10560
MSM10570
MSM10580
MSM10590
MSM10600
MSM10610
MSM10620
MSM10630
MSM10640
MSM10650
MSM10660
MSM10670
MSM10680
MSM10690
MSM10700
MSM10710
MSM10720
MSM10730
MSM10740
MSM10750
MSM10760
MSM10770
MSM10780
MSM10790
MSM10800
MSM10810
MSM10820
MSM10830
MSM10840
MSM10850
MSM10860
MSM10870
MSM10880

msml.for

```
      WRITE (6,*) 'AUGPAR PARAMETERS'
      WRITE(6,810)
810  FORMAT(1X,' IYEAR',2X,' IDAY',2X,' IHOURL',2X,' IMIN',
2      2X,' ISEC',2X,' KP',2X,' DST',2X,' EQEDGE',
3      2X,' DLATAZ',2X,' DLATRV',2X,' MLTRV')
      WRITE(6,820) (AUGPAR(LM,IT),LM=1,11)
820  FORMAT(1X,11(F8.2,2X))
      WRITE(6,830)
830  FORMAT(1X)
      WRITE(6,840)
840  FORMAT(1X,' IMFBX',2X,' IMFBI',2X,' IMFBZ',2X,' CLAPSE',
2      2X,' GEOMGX',2X,' GEOMGI',2X,' GEOMGZ',2X,' GEOMGT',
3      2X,' LBOUND',2X,' SWVEL',2X,' SWDEN')
      WRITE(6,820) (AUGPAR(LM,IT),LM=12,22)
      WRITE(6,830)
      WRITE(6,850)
850  FORMAT(1X,' TILTW',2X,' STAND',2X,' PCP',2X,' IPATT',
2      2X,' DDST',2X,' DEQDT')
      WRITE(6,820) (AUGPAR(LM,IT),LM=23,28)
      WRITE(6,830)

C
C
C
C
C      GET THE CURRENT TIME
C
C      CALL DTIME(ISTBT)
C
C      CALCULATE APPROXIMATE PLASMAPAUSE RADIUS FOR USE IN PPTM FOR
C      DETERMINING WEAK LOSS RATE
      RPP(L)=1./(COS(AUGPAR(EQEDGE,L)*PI/180.))**2
C
C
C      CALCULATE B FIELD MATRICES
C
      FSTOFF=AUGPAR(STAND,L)
      FEQEDG=AUGPAR(EQEDGE,L)
      FDST=AUGPAR(DST,L)
      FCLPSE=AUGPAR(CLAPSE,L)
      FTILT=AUGPAR(TILTW,L)
C
      WRITE (6,*) 'L =',L,' FSTOFF =',FSTOFF,' FEQEDG =',FEQEDG,
1      ' FDST =',FDST,' FCLPSE =',FCLPSE,' FTILT =',
2      FTILT,' RPP =',RPP(L)
C
      CALL BTRACE(LATDIM,LTDIM,ITMDIM,L,FSTOFF,FEQEDG,FDST,
2      FCLPSE,FTILT,VM,BMIN,XMIN,YMIN,ZMIN,R,P,ALOC,TETA)
C
C      GET THE CURRENT TIME AND FIND THE TIME SPENT IN BTRACE
C
C      CALL DTIME(IENDBT)
      WRITE (LUERR,*) 'THE TIME SPENT IN BTRACE IS ',
1      ABS(IENDBT-ISTBT)/100.,' SECONDS.'
C
      WRITE(6,*) 'MY UT =',(ITIME(ISECND)/3600)*100+
1      MOD(ITIME(ISECND),3600)/60,' DAY= ',ITIME(IDAY)
      WRITE(6,*) 'L=',L
C
      ICNTRL = 1
C
C      GET THE CURRENT TIME
C
C      CALL DTIME(ISTEF)
C
```

MSM10890
MSM10900
MSM10910
MSM10920
MSM10930
MSM10940
MSM10950
MSM10960
MSM10970
MSM10980
MSM10990
MSM11000
MSM11010
MSM11020
MSM11030
MSM11040
MSM11050
MSM11060
MSM11070
MSM11080
MSM11090
MSM11100
MSM11110
MSM11120
MSM11130
MSM11140
MSM11150
MSM11160
MSM11170
MSM11180
MSM11190
MSM11200
MSM11210
MSM11220
MSM11230
MSM11240
MSM11250
MSM11260
MSM11270
MSM11280
MSM11290
MSM11300
MSM11310
MSM11320
MSM11330
MSM11340
MSM11350
MSM11360
MSM11370
MSM11380
MSM11390
MSM11400
MSM11410
MSM11420
MSM11430
MSM11440
MSM11450
MSM11460
MSM11470
MSM11480
MSM11490
MSM11500
MSM11510
MSM11520

```
C
WRITE(6,*) 'IN EBCON'
WRITE(6,*) 'EQEDGE= ',AUGPAR(EQEDGE,L), ' STAND= ',AUGPAR(STAND,L)
C GET BNDYS FOR USE IN EFIELD
VDROP=AUGPAR(PCP,L)*1.E3
CALL EFBNDY (LATDIM,LTDIM,ITMDIM,L,AUGPAR(EQEDGE,L),
2 AUGPAR(STAND,L),VDROP,R,ALOCT,COLAT,
3 ASOUTH(1,L),BSOUTH(1,L),DXS(1,L),DYS(1,L))
CALL EFBNDY (LATDIM,LTDIM,ITMDIM,L,AUGPAR(EQEDGE,L),
2 AUGPAR(STAND,L),VDROP,R,ALOCT,COLAT,
3 ANORTH(1,L),BNORTH(1,L),DXN(1,L),DYN(1,L))
WRITE(6,*) 'AFTER EFBNDY L= ',L, ' A= ', (ANORTH(IIC,L),IIC=1,3)
WRITE(6,*) ' BNORTH= ', (BNORTH(IIC,L),IIC=1,3)
WRITE(6,*) ' DXN= ', (DXN(IIC,L),IIC=1,3)
WRITE(6,*) ' DYN= ', (DYN(IIC,L),IIC=1,3)
C
C
C
IPATRN=AUGPAR(IPATT,L)
CALL EFIELD(IPATRN,VDROP,AUGPAR(DEQDT,L),
2 ASOUTH,BSOUTH,DXS,DYS,ANORTH,BNORTH,DXN,DYN,COLAT,
3 ALOCT,MODE,LATDIM,LTDIM,JWRAP,ITMDIM,L,ICNTRL,1,V,
4 VNORTH,VSOUTH)
C
C GET THE CURRENT TIME AND FIND THE TIME SPENT IN EFIELD
C
C CALL DTIME(IENDEF)
C WRITE (LUERR,*) 'THE TIME SPENT IN EFIELD IS ',
C 1 ABS(IENDEF-ISTEF)/100., ' SECONDS.'
C
C ***** WRITE V
C
C CALL OUTP(V(1,1,L),LATDIM,LTDIM,1,LATDIM,1,1,LTDIM,1,0.,
1 'V',6,132)
WRITE(6,*) 'MY UT =', (ITIME(ISECND)/3600)*100+
1 MOD(ITIME(ISECND),3600)/60, ' DAY= ',ITIME(IDAY)
WRITE(6,*) 'L=',L
C
C CALL OUTP(VNORTH,LATDIM,LTDIM,1,LATDIM,1,1,LTDIM,1,0.,
C 2 'VNORTH',6,132)
C CALL OUTP(VSOUTH,LATDIM,LTDIM,1,LATDIM,1,1,LTDIM,1,0.,
C 2 'VSOUTH',6,132)
C
C
C FIND BNDLOC FOR THIS TIME
C
C CALL FNDBND(L,COLAT,ALOCT,ANORTH(2,L),BNORTH(2,L),
1 DXN(2,L),DYN(2,L),LATDIM,LTDIM,ITMDIM,BNDLOC)
C
C COMPUTE RBNDY
WRITE(6,*) 'L= ',L
DO 8 J=1,LTDIM
BI=BNDLOC(J,L)
BJ=J
BL=L
RBNDY=G3NTRP(R,LATDIM,LTDIM,ITMDIM,BI,BJ,BL)
WRITE(6,*) 'J= ',J, ' BNDLOC= ',BNDLOC(J,L), ' RBNDY= ',RBNDY
8 CONTINUE
C
C
C
IBM=LATDIM
DO 6 JJ=1,LTDIM
```

MSM11530
MSM11540
MSM11550
MSM11560
MSM11570
MSM11580
MSM11590
MSM11600
MSM11610
MSM11620
MSM11630
MSM11640
MSM11650
MSM11660
MSM11670
MSM11680
MSM11690
MSM11700
MSM11710
MSM11720
MSM11730
MSM11740
MSM11750
MSM11760
MSM11770
MSM11780
MSM11790
MSM11800
MSM11810
MSM11820
MSM11830
MSM11840
MSM11850
MSM11860
MSM11870
MSM11880
MSM11890
MSM11900
MSM11910
MSM11920
MSM11930
MSM11940
MSM11950
MSM11960
MSM11970
MSM11980
MSM11990
MSM12000
MSM12010
MSM12020
MSM12030
MSM12040
MSM12050
MSM12060
MSM12070
MSM12080
MSM12090
MSM12100
MSM12110
MSM12120
MSM12130
MSM12140
MSM12150
MSM12160

90/06/30
12:03:44

20

msm1.for

```
IF (INT (BNDLOC (JJ, L)) .LT. IBM) THEN
    IBM=INT (BNDLOC (JJ, L))
END IF
6 CONTINUE

C
C
C ***** WRITE VM, BMIN, XMIN, YMIN, ZMIN
C
WRITE (6, *) 'MY UT =', (ITIME (ISECND) / 3600) * 100 +
1 MOD (ITIME (ISECND), 3600) / 60
WRITE (6, *) 'L=', L
    CALL OUTP (VM (1, 1, L), LATDIM, LTDIM, IBM, LATDIM, 1, 3, LTDIM-1, 1,
1 1., 'VM', 6, 132)
    CALL OUTP (BMIN, LATDIM, LTDIM, IBM, LATDIM, 2, 3, LTDIM-3, 3, 10.,
C 1 'BMIN', 6, 132)
C WRITE (6, *) 'MY UT =', (ITIME (ISECND) / 3600) * 100 +
C 1 MOD (ITIME (ISECND), 3600) / 60, ' DAY= ', ITIME (IDAY)
C WRITE (6, *) 'L=', L
C    CALL OUTP (XMIN, LATDIM, LTDIM, IBM, LATDIM, 2, 3, LTDIM-3, 3, 1.,
C 1 'XMIN', 6, 132)
C WRITE (6, *) 'MY UT =', (ITIME (ISECND) / 3600) * 100 +
C 1 MOD (ITIME (ISECND), 3600) / 60, ' DAY= ', ITIME (IDAY)
C WRITE (6, *) 'L=', L
C    CALL OUTP (YMIN, LATDIM, LTDIM, IBM, LATDIM, 2, 3, LTDIM-3, 3, 1.,
C 1 'YMIN', 6, 132)
C WRITE (6, *) 'MY UT =', (ITIME (ISECND) / 3600) * 100 +
C 1 MOD (ITIME (ISECND), 3600) / 60, ' DAY= ', ITIME (IDAY)
C WRITE (6, *) 'L=', L
C    CALL OUTP (ZMIN, LATDIM, LTDIM, IBM, LATDIM, 2, 3, LTDIM-3, 3, 1.,
C 1 'ZMIN', 6, 132)
C WRITE (6, *) 'MY UT =', (ITIME (ISECND) / 3600) * 100 +
C 1 MOD (ITIME (ISECND), 3600) / 60, ' DAY= ', ITIME (IDAY)
C WRITE (6, *) 'L=', L
C    CALL OUTP (R (1, 1, L), LATDIM, LTDIM, IBM, 45, 1, 1, LTDIM, 1, 1.,
C 1 'R', 6, 132)
C
C WRITE (6, *) 'MY UT =', (ITIME (ISECND) / 3600) * 100 +
C 1 MOD (ITIME (ISECND), 3600) / 60, ' DAY= ', ITIME (IDAY)
C WRITE (6, *) 'L=', L
C    CALL OUTP (P (1, 1, L), LATDIM, LTDIM, IBM, LATDIM, 1, 1, LTDIM, 1, 0.01,
C 1 'P', 6, 132)
C SET VMLOSS FOR THE CURRENT VM PARAMETERS
C
C    CALL VMLSET (L, VM, BIR, SINI, LATDIM, LTDIM, ITMDIM, VMLOSS)
C
C ***** WRITE VMLOSS
C
C    CALL OUTP (VMLOSS (1, 1, L), LATDIM, LTDIM, IBM, LATDIM, 2, 3, LTDIM-3,
C 1 3, 0.01, 'VMLOSS', 6, 132)
C
C SET UP ID ARRAYS
C
C    WRITE (6, *) 'BEFORE ID ARRAY LOADED L=', L
C        ID (1) = ITIME (IYEAR)
C        ID (2) = ITIME (IDAY)
C        ID (3) = ITIME (ISECND) / 3600
C        ID (4) = MOD (ITIME (ISECND), 3600) / 60
C        ID (5) = MOD (ITIME (ISECND), 60)
C        ID (11) = L
C        RID (1) = TIMTAG (L)
C        RID (2) = AUGPAR (KP, L)
C        RID (3) = AUGPAR (PCP, L)
C        RID (4) = AUGPAR (DEQDT, L)
```

MSM12170
MSM12180
MSM12190
MSM12200
MSM12210
MSM12220
MSM12230
MSM12240
MSM12250
MSM12260
MSM12270
MSM12280
MSM12290
MSM12300
MSM12310
MSM12320
MSM12330
MSM12340
MSM12350
MSM12360
MSM12370
MSM12380
MSM12390
MSM12400
MSM12410
MSM12420
MSM12430
MSM12440
MSM12450
MSM12460
MSM12470
MSM12480
MSM12490
MSM12500
MSM12510
MSM12520
MSM12530
MSM12540
MSM12550
MSM12560
MSM12570
MSM12580
MSM12590
MSM12600
MSM12610
MSM12620
MSM12630
MSM12640
MSM12650
MSM12660
MSM12670
MSM12680
MSM12690
MSM12700
MSM12710
MSM12720
MSM12730
MSM12740
MSM12750
MSM12760
MSM12770
MSM12780
MSM12790
MSM12800

90/06/30
12:03:44

msm1.for

21

```
C
C -----> WRITE AUGPAR TO AN ARCHIVAL FILE *****
      ID(8)=NAUGEL
      ID(9)=1
      ID(10)=ITMMAX
      CALL WRT3D(LUPDAT, IRECEB, ID, RID, CHID, AUGPAR(1, L), NAUGEL,
2         1, 1, ITMDIM)
C
C -----> WRITE VM, BMIN, XMIN, YMIN, ZMIN, AND V TO AN ARCHIVAL FILE
C
C      ID(8)=LATDIM
C      ID(9)=LTDIM
C      ID(10)=1
C
      CALL WRT3D(LUBFLD, IRECEB, ID, RID, CHID, VM(1, 1, L),
1         LATDIM, LTDIM, 1, ITMDIM)
      CALL WRT3D(LUBMIN, IRECEB, ID, RID, CHID, BMIN,
1         LATDIM, LTDIM, 1, ITMDIM)
      CALL WRT3D(LUXMIN, IRECEB, ID, RID, CHID, XMIN, LATDIM,
1         LTDIM, 1, ITMDIM)
      CALL WRT3D(LUYMIN, IRECEB, ID, RID, CHID, YMIN, LATDIM,
1         LTDIM, 1, ITMDIM)
      CALL WRT3D(LUZMIN, IRECEB, ID, RID, CHID, ZMIN, LATDIM,
1         LTDIM, 1, ITMDIM)
      CALL WRT3D(LUEFLD, IRECEB, ID, RID, CHID, V(1, 1, L), LATDIM,
1         LTDIM, 1, ITMDIM)
      CALL WRT3D(LUVN, IRECEB, ID, RID, CHID, VNORTH, LATDIM,
1         LTDIM, 1, ITMDIM)
      CALL WRT3D(LUVS, IRECEB, ID, RID, CHID, VSOUTH, LATDIM,
1         LTDIM, 1, ITMDIM)
C
C ----> WRITE ALOCT, COLAT, AND BNDLOC TO ARCHIVAL FILE
C
      ID(8)=LATDIM
      ID(9)=LTDIM
      ID(10)=1
C
      CALL WRT3D(LUALOC, IRECEB, ID, RID, CHID, ALOCT, LATDIM,
1         LTDIM, 1, ITMDIM)
      CALL WRT3D(LUCOLT, IRECEB, ID, RID, CHID, COLAT, LATDIM,
1         LTDIM, 1, ITMDIM)
C
      ID(8)=LTDIM
      ID(9)=1
      ID(10)=1
C
      CALL WRT3D(LUBNDL, IRECEB, ID, RID, CHID, BNDLOC(1, L),
1         LTDIM, 1, 1, ITMDIM)
C
C
C INCREMENT ITIME
C
12 CONTINUE
C
      CALL TIMINC(ITYME, ITMNEW, IINC)
C
      ITIME(IYEAR)=ITMNEW(IYEAR)
      ITIME(IDAY)=ITMNEW(IDAY)
      ITIME(ISECND)=ITMNEW(ISECND)
```

MSM12810
MSM12820
MSM12830
MSM12840
MSM12850
MSM12860
MSM12870
MSM12880
MSM12890
MSM12900
MSM12910
MSM12920
MSM12930
MSM12940
MSM12950
MSM12960
MSM12970
MSM12980
MSM12990
MSM13000
MSM13010
MSM13020
MSM13030
MSM13040
MSM13050
MSM13060
MSM13070
MSM13080
MSM13090
MSM13100
MSM13110
MSM13120
MSM13130
MSM13140
MSM13150
MSM13160
MSM13170
MSM13180
MSM13190
MSM13200
MSM13210
MSM13220
MSM13230
MSM13240
MSM13250
MSM13260
MSM13270
MSM13280
MSM13290
MSM13300
MSM13310
MSM13320
MSM13330
MSM13340
MSM13350
MSM13360
MSM13370
MSM13380
MSM13390
MSM13400
MSM13410
MSM13420
MSM13430
MSM13440

90/06/30
12:03:44

msml.for

22

10 CONTINUE

C
C
C

RETURN
END

C
C
C
C

***** B L O C K D A T A *****

BLOCK DATA

C
C
C
C
C
C
C
C

VERSION 1.0 DATE: JANUARY 9, 1988

PROGRAMMER: BRYAN A. BALES

PURPOSE: THIS BLOCK DATA CONTAINS THE INDEX OF DATA ELEMENTS IN THE
PARRAY AND AUGPAR ARRAYS.

INTEGER IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST, DDST,
1 EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2 GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3 STAND, PCP, DEQDT, IPATT

C

COMMON /ARINDX/IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST,
1 EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2 GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3 STAND, PCP, IPATT, DDST, DEQDT

C

COMMON /LUNIT/LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,
1 LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,
2 LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,
3 LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH

C

COMMON /GRID/DLAM, DPSI, RI, JWRAP

C

COMMON /MOVE/IBEG, ISTOP, JBEG, JSTOP, IEBEG, IEEND

C

DATA IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST,
1 EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2 GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3 STAND, PCP, IPATT, DDST, DEQDT
4 /1, 2, 3, 3, 4, 5, 6, 7,
5 8, 9, 10, 11, 12, 13, 14, 15,
6 16, 17, 18, 19, 20, 21, 22, 23,
7 24, 25, 26, 27, 28/

C

DATA LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,
1 LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,
2 LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,
3 LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH
4 /9, 13, 14, 15, 16, 17,
5 20, 21, 22, 23, 24, 25, 26, 27,
6 28, 35, 36, 37, 38, 39, 40, 41,
7 42, 43, 44, 45, 46, 47, 48, 49/

DATA IBEG, ISTOP, JBEG, JSTOP, IEBEG /1, 53, 3, 50, 1/

C

END

C

SUBROUTINE EMODEL(IPATT, PCP, DEQDT, A, B, DX, DY, COLAT, ALOCT, MODE,
1 LATDIM, LTDIM, JWRAP, ICNTRL, ITOP, V)

C

C

THIS SUBROUTINE RETURNS VALUES OF THE POTENTIAL V AT ALL GRID

MSM13450
MSM13460
MSM13470
MSM13480
MSM13490
MSM13500
MSM13510
MSM13520
MSM13530
MSM13540
MSM13550
MSM13560
MSM13570
MSM13580
MSM13590
MSM13600
MSM13610
MSM13620
MSM13630
MSM13640
MSM13650
MSM13660
MSM13670
MSM13680
MSM13690
MSM13700
MSM13710
MSM13720
MSM13730
MSM13740
MSM13750
MSM13760
MSM13770
MSM13780
MSM13790
MSM13800
MSM13810
MSM13820
MSM13830
MSM13840
MSM13850
MSM13860
MSM13870
MSM13880
MSM13890
MSM13900
MSM13910
MSM13920
MSM13930
MSM13940
MSM13950
MSM13960
MSM13970
MSM13980
MSM13990
MSM14000
MSM14010
MSM14020
MSM14030
MSM14040
MSM14050
MSM14060
MSM14070
MSM14080

90/06/30
12:03:44

msml.for

23

POINTS, SPECIFIED BY COLAT(I,J) AND ALOCT(I,J).

REFERENCE: SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-87-K-0001.

GENERAL CALL PARAMETERS FOR SUBROUTINE EMODEL

IPATT = HEPPNER-MAYNARD PATTERN NO. MUST BE SPECIFIED AS AN INTEGER BETWEEN 1 AND 7.

IPATT=1 MEANS PATTERN A, FOR BZ SOUTH, BY=0.

IPATT=2 MEANS PATTERN BC, FOR BZ SOUTH, BYNORTH>0

IPATT=3 MEANS PATTERN DE, FOR BZ SOUTH, BYNORTH<0

IPATT=4 MEANS PATTERN BCP, TWISTED BC, FOR BZ>0

IPATT=5 MEANS PATTERN BCPP, TWISTED BC, BZ STRONG>0

IPATT=6 MEANS PATT. DEP, TWISTED DE, BZ WEAK >0

IPATT=7 MEANS PATT. DEPP, TWISTED DE, BZ STRONG>0

PCP = POLAR-CAP POTENTIAL DROP IN VOLTS. MUST BE SPECIFIED UNLESS USING UNSCALED HEPPNER-MAYNARD POTENTIAL(ICNTRL=-1)

DEQDT = ESTIMATED D/DT OF LATITUDE OF EQUATORWARD EDGE OF AURORAL ZONE AT LOCAL MIDNIGHT, IN DEGREES/HOUR. THIS IS A PARAMETER IN LOW-LATITUDE E-FIELD MODEL. NOT NEEDED IF ICNTRL.LE.0.

VECTORS A(3), B(3), DX(3), DY(3) DESCRIBE THE ELLIPSES THAT FORM THE BOUNDARIES BETWEEN REGIONS 1, 2, AND 3. BOUNDARY 1 IS THE EQUATORWARD EDGE OF THE POLAR CAP. BOUNDARY 2 IS THE EQUATORWARD EDGE OF REGION 1 OR THE EQUATORWARD EDGE OF THE FIELD-REVERSAL REGION. BOUNDARY 3 IS THE SHIELDING LAYER.

A(L) = RADIUS OF ELLIPSE MEASURED IN X(SUNWARD) DIRECTION.

B(L) = RADIUS OF ELLIPSE MEASURED IN Y(DUSKWARD) DIRECTION.

DX(L) = SUNWARD DISPLACEMENT OF COORD. SYSTEM CENTER FROM POLE.

DY(L) = DUSKWARD DISPL. OF COORD. SYSTEM CENTER FROM POLE.

TO USE EFIELD WITH RCM, CHOOSE A(2) = B(2) = TETA(IMIN),

DX(2) = -OFFSET*180./PI, DY(2) = 0. ELLIPSE PARAMETERS FOR BOUNDARIES 1 AND 3 NEED NOT BE SPECIFIED.

COLAT AND ALOCT ARE THE USUAL MAGNETIC COLATITUDE AND MAGNETIC LOCAL TIME ANGLES IN RADIANs.

MODE IS A DUMMY PARAMETER, NOT CURRENTLY USED.

IMAX, JMAX, JWRAP HAVE THEIR USUAL MEANING, SAME AS IN RCM.

ITMDIM = NUMBER OF TIME LABELS IN THE V-MATRIX. FOR USE WITH RCM, IT SHOULD BE SET EQUAL TO 1.

ITMCUR = TIME-LABEL NUMBER OF V-MATRIX CURRENTLY BEING COMPUTED. FOR USE WITH RCM, IT SHOULD BE SET EQUAL TO 1.

ICNTRL = 1 MEANS DIFFERENT FORMULAS ARE USED IN REGIONS 1,2,3.

= 0 MEANS HEPPNER-MAYNARD FORMULA IS USED FOR ALL

LATITUDES, BUT SCALED TO EXTERNALLY SPECIFIED PCP AND ELLIPSE PARAMETERS.

= -1 MEANS HEPPNER-MAYNARD FORMULA IS USED FOR ALL LATITUDES, UNSCALED.

= -2 IS THE MODE USED FOR USE WITH RCM. HEPPNER-MAYNARD IS USED POLEWARD OF ELLIPSE 2, AND ON ELLIPSE 2, SCALED TO FIT SPECIFIED POTENTIAL DROP AND DIMENSIONS OF ELLIPSE 2.

MSM14090
MSM14100
MSM14110
MSM14120
MSM14130
MSM14140
MSM14150
MSM14160
MSM14170
MSM14180
MSM14190
MSM14200
MSM14210
MSM14220
MSM14230
MSM14240
MSM14250
MSM14260
MSM14270
MSM14280
MSM14290
MSM14300
MSM14310
MSM14320
MSM14330
MSM14340
MSM14350
MSM14360
MSM14370
MSM14380
MSM14390
MSM14400
MSM14410
MSM14420
MSM14430
MSM14440
MSM14450
MSM14460
MSM14470
MSM14480
MSM14490
MSM14500
MSM14510
MSM14520
MSM14530
MSM14540
MSM14550
MSM14560
MSM14570
MSM14580
MSM14590
MSM14600
MSM14610
MSM14620
MSM14630
MSM14640
MSM14650
MSM14660
MSM14670
MSM14680
MSM14690
MSM14700
MSM14710
MSM14720

90/06/30
12:03:44

msm1.for

24

```
C
C      ITOP = 1 MEANS THAT A(1), B(1), DX(1), DY(1) ARE COMPUTED
C      INTERNALLY IN EFIELD. OTHERWISE THEY MUST BE PASSED TO
C      EFIELD FROM OUTSIDE. ITOP IS NORMALLY = 1 FOR USE IN
C
C      MSM. ITOP SHOULD BE SET TO 1 FOR USE WITH RCM.
C      V = POTENTIAL MATRIX COMPUTED IN EMODEL. IT IS DIMENSIONED AT
C      LATDIM X LTDIM
C
C      SUBROUTINE EMODEL CALLS SUBROUTINES EPOT(HEPPNER-MAYNARD),
C      LOW(LOW-LATITUDES), AURORA(REGION 2), REG1(REGION 1).
C
C
C      PARAMETER(JDIM=51)
C      DIMENSION THETAA(JDIM), THETAB(JDIM), THETAC(JDIM)
C      DIMENSION COLAT(LATDIM, LTDIM), ALOCT(LATDIM, LTDIM)
C      DIMENSION V(LATDIM, LTDIM)
C      DIMENSION VA(JDIM), DVADT(JDIM), VC(JDIM), DVCDT(JDIM)
C      DIMENSION VB(JDIM), VBLOW(JDIM), VBAZ(JDIM), DVBLO(JDIM)
C      DIMENSION DVBDT(JDIM), DVBAZ(JDIM)
C      DIMENSION COEF(18,18), MLBL(5)
C      DIMENSION A(3), B(3), DX(3), DY(3)
C      DIMENSION XCO(18,18), AHM(2,7), BHM(2,7), DXHM(2,7)
C      DIMENSION DYHM(2,7), VMIN(7), VMAX(7)
C
C      PARAMETER VALUES
C
C      NTAPE = FILE NUMBER FROM WHICH MAYNARD-RICH COEFFICIENTS ARE
C      READ.
C
C      IMAX = LATDIM
C      JMAX = LTDIM
C      IF(JDIM.NE.LTDIM) THEN
184      WRITE(6,184) JDIM, LTDIM
      FORMAT(1H, 'JDIM = ', I4, ' LTDIM = ', I4, ' IN EFIELD.')
      END IF
C
C      WRITE(6,*) 'IPATT,PCP,DEQDT', IPATT,PCP,DEQDT
C
C      PI = 3.141593
C      PI2 = 2.*PI
C      NTAPE = 7
C      NNMAX = 16
C      EPS = .01
C      G = 1.
C      IEFF = IMAX
C      CALL INPUT(NTAPE, COEF, IPATT, XCO, AHM, BHM, DXHM, DYHM, VMIN,
1      VMAX)
C
C      FG = SCALING FACTOR THAT CONVERTS HM POTENTIAL TO PRESENT SITUATION
C
C      IF(ICNTRL.GE.0) THEN
C      FG = PCP/(VMAX(IPATT)-VMIN(IPATT))
C      ELSE
C      FG = 1.
C      END IF
C      WRITE(6,903) IPATT, VMAX(IPATT), VMIN(IPATT), FG
903  FORMAT(1H, 'IPATT, VMAX, VMIN, FG = ', I5, 3E15.5)
C
C      COMPUTE A(1), B(1), DX(1), DY(1) IF ITOP = 1.
```

```
C
      IF (ITOP.EQ.1) THEN
        A(1) = A(2)*AHM(1,IPATT)/AHM(2,IPATT)
        B(1) = B(2)*BHM(1,IPATT)/BHM(2,IPATT)
        DX(1) = DX(2) + A(2)*(DXHM(1,IPATT)-DXHM(2,IPATT))/AHM(2,IPATT)
        DY(1) = DY(2) + B(2)*(DYHM(1,IPATT)-DYHM(2,IPATT))/BHM(2,IPATT)
      END IF
C
      VMAXX = VMAX(IPATT)*FG
      VMINN = VMIN(IPATT)*FG
      SA=SIN((A(3)-DX(3))*PI/180.)
      VPENET = -13112.*G*DEQDT*SA
      VBAR = 0.6*VMAXX + 0.4*VMINN -0.22*VPENET
C
      IF (ICNTRL.NE.-2) THEN
        WRITE(6,905) IPATT,VMAXX,VMINN
905      FORMAT(1H,'IPATT,VMAXX,VMINN = ',I5,2E15.5)
        WRITE(6,906) A(1),A(2),A(3)
906      FORMAT(1H,'A(1), A(2), A(3) = ',3E15.5)
        WRITE(6,907) B(1),B(2),B(3)
907      FORMAT(1H,'B(1), B(2), B(3) = ',3E15.5)
        WRITE(6,908) DX(1), DX(2), DX(3)
908      FORMAT(1H,'DX(1), DX(2), DX(3) = ',3E15.5)
        WRITE(6,909) DY(1), DY(2), DY(3)
909      FORMAT(1H,'DY(1), DY(2), DY(3) = ',3E15.5)
        WRITE(6,899) VPENET,VBAR,VMAXX,VMINN
899      FORMAT(1H,'VPENET,VBAR,VMAX,VMIN=',4E15.5)
C
      END IF
C
C
      IF (ICNTRL.EQ.0.OR.ICNTRL.EQ.-1) GO TO 15
C
      COMPUTATION OF BOUNDARY LOCATIONS AS FCNS OF J.
C
      GUARD AGAINST IRRELEVANT A'S OR B'S BEING ZERO OR NEGATIVE.
C
      DO 811 L=1,3
        IF (A(L).LT.0.001) A(L)=0.001
        IF (B(L).LT.0.001) B(L) = 0.001
811  CONTINUE
C
C
      DO 1 J = 1, JMAX
        AL = ALOCT(IEFF,J)
        THETAA(J) = THET(A(1),B(1),DX(1),DY(1),AL)*PI/180.
        THETAB(J) = THET(A(2),B(2),DX(2),DY(2),AL)*PI/180.
        THETAC(J) = THET(A(3),B(3),DX(3),DY(3),AL)*PI/180.
1      CONTINUE
C
C
      SEGMENT FOR DEFINING BOUNDARY POTENTIALS AND E-FIELDS
C
      DO 7 J=1, JMAX
C
      POTENTIAL AND DERIVATIVE AT BOUNDARY A
C
        GLONG = ALOCT(IEFF,J) + PI
        IF (GLONG.GT.(2.*PI)) GLONG = GLONG - 2.*PI
        GMLAT = 90. - THETAA(J)*180./PI
        CALL EPOT(COEF,GMLAT,GLONG,VA(J),NTAPE,IPATT,NNMAX,A,B,
1      DX,DY,XCO,AHM,BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)
        GMLAT = GMLAT - EPS*180./PI
        CALL EPOT(COEF,GMLAT,GLONG,VT,NTAPE,IPATT,NNMAX,A,B,
1      DX,DY,XCO,AHM,BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)
```

90/06/30
12:03:44

msm1.for

26

	DVADT(J) = (VT-VA(J))/EPS	MSM16010
7	CONTINUE	MSM16020
C		MSM16030
C	POTENTIAL AND DERIVATIVE AT C	MSM16040
C		MSM16050
	DO 8 J=1, JMAX	MSM16060
	CALL LOW(THETAC(J), ALOCT(IEFF,J), VC(J), DEQDT,A(3), DX(3), VBAR,G)	MSM16070
	TT = THETAC(J) + EPS	MSM16080
	CALL LOW(TT, ALOCT(IEFF,J), VT, DEQDT,A(3), DX(3), VBAR,G)	MSM16090
	DVCDT(J) = (VT-VC(J))/EPS	MSM1610C
8	CONTINUE	MSM16110
C		MSM16120
C		MSM16130
C	POTENTIAL AT B	MSM16140
C		MSM16150
	DO 9 J=1, JMAX	MSM16160
	GLONG = ALOCT(IEFF,J)+PI	MSM16170
	IF(GLONG.GT.(2.*PI)) GLONG = GLONG-2.*PI	MSM16180
	GMLAT = 90. - THETAB(J)*180./PI	MSM16190
	CALL EPOT(COEF,GMLAT,GLONG,VB(J),NTAPE,IPATT,NNMAX,A,B,	MSM16200
1	DX,DY,XCO,AHM,BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)	MSM16210
	VBLOW(J) = VC(J) - DVCDT(J)*(THETAC(J)-THETAB(J))	MSM16220
	VBAZ(J) = VB(J) - VBLOW(J)	MSM16230
9	CONTINUE	MSM16240
C		MSM16250
C		MSM16260
C	DERIVATIVE AT POTENTIAL AT B	MSM16270
C		MSM16280
	DO 10 J = 1, JMAX	MSM16290
	DVBLO(J) = DVCDT(J)	MSM16300
	TT = THETAB(J) + EPS	MSM16310
	CALL AURORA(TT,ALOCT(IEFF,J),VT,VBAZ,THETAB(J),THETAC(J),J,	MSM16320
1	JMAX,JWRAP)	MSM16330
	DVBAZ(J) = (VT-VBAZ(J))/EPS	MSM16340
	DVBBDT(J) = DVBLO(J) + DVBAZ(J)	MSM16350
10	CONTINUE	MSM16360
	NNMAX=NNMAX	MSM16370
C		MSM16380
C	MAIN DO LOOP OVER GRID	MSM16390
C		MSM16400
15	CONTINUE	MSM16410
	DO 5 I = 1, IMAX	MSM16420
	DO 5 J = 1, JMAX	MSM16430
	V(I,J) = 0.	MSM16440
C		MSM16450
C		MSM16460
C		MSM16470
C	CALLS OF DIFFERENT SUBROUTINE SEQUENCES	MSM16480
C		MSM16490
C	REGION ZERO = POLAR CAP	MSM16500
C		MSM16510
	IF(COLAT(I,J).LE.THETAA(J).OR.ICNTRL.EQ.0.OR.ICNTRL.EQ.-1) THEN	MSM16520
	GLONG = ALOCT(I,J) + PI	MSM16530
	IF(GLONG.GT.PI2) GLONG=GLONG-PI2	MSM16540
	GMLAT = 90. - COLAT(I,J)*180./PI	MSM16550
	CALL EPOT(COEF,GMLAT,GLONG,VTEMP,NTAPE,IPATT,NNMAX,A,	MSM16560
1	B,DX,DY,XCO,AHM,BHM,DXHM,DYHM,VMIN,VMAX,PCP,ICNTRL)	MSM16570
	END IF	MSM16580
C		MSM16590
C	REGION 3 = LOW LATITUDES	MSM16600
C		MSM16610
	IF(COLAT(I,J).GE.THETAC(J).AND.ICNTRL.GT.0) THEN	MSM16620
	CALL LOW(COLAT(I,J),ALOCT(I,J),VTEMP,DEQDT,A(3),DX(3),VBAR,G)	MSM16630
C		MSM16640

90/06/30
12:03:44

msm1.for

27

```

C      END IF
C
C      REGION 2 = LOWER AURORAL ZONE
C
C      IF (COLAT(I,J) .LT. THETAC(J) .AND. COLAT(I,J) .GE. THETAB(J) .AND. ICNTRL
1  GT.0) THEN
C          VLOW = VC(J) + DVCDT(J) * (COLAT(I,J) - THETAC(J))
C          CALL AURORA(COLAT(I,J), ALOCT(I,J), VAZ, VBAZ, THETAB(J), THETAC(J),
%      J, JMAX, JWRAP)
C          VTEMP = VAZ + VLOW
C      END IF
C
C      IF (COLAT(I,J) .LT. THETAB(J) .AND. COLAT(I,J) .GT. THETAA(J) .AND. ICNTRL
1  GT.0) THEN
C          CALL REG1(COLAT(I,J), ALOCT(I,J), VTEMP, THETAA(J),
%      THETAB(J), VA(J), VB(J), DVADT(J), DVBDT(J))
C      END IF
C
C      USE HEPPNER-MAYNARD DIRECTLY IN REGION 1 FOR USE WITH RCM.
C
C      TB = THETAB(J) + 1.E-4
C      IF (COLAT(I,J) .LT. TB .AND. COLAT(I,J) .GT. THETAA(J) .AND. ICNTRL.EQ.-2)
1  THEN
C          GLONG = ALOCT(I,J) + PI
C          IF (GLONG.GT.PI2) GLONG = GLONG - PI2
C          CALL EPOT(COEF, GMLAT, GLONG, VTEMP, NTAPE, IPATT, NNMAX, A,
1      B, DX, DY, XCO, AHM, BHM, DXHM, DYHM, VMIN, VMAX, PCP, ICNTRL)
C      END IF
C
C      V(I,J) = VTEMP
C
C      5  CONTINUE
C      RETURN
C      END
C
C
C
C
C
C
C      SUBROUTINE EPOT(COEF, TLAT, TLON, VALUE, NTAPE, IPATT, NNMAX, A, B,
1  DX, DY, XCO, AHM, BHM, DXHM, DYHM, VMIN, VMAX, PCP, ICNTRL)
C
C  PURPOSE:  THIS SUBROUTINE USES THE SCALED HEPPNER-MAYNARD MODEL TO
C            RETURN THE ELECTRIC POTENTIAL AT A SPECIFIC POINT.
C
C  REFERENCE: SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-
C            87-K-0001.
C
C  INPUTS:
C      COEF      COEFFICIENTS FROM DIGITIZATION OF HEPPNER-MAYNARD
C                E-FIELD MODEL
C      TLAT      COLATITUDE OF PT WHERE V IS TO BE CALCULATED
C                (RADIAN)
C      TLON      MLT OF PT WHERE V IS TO BE CALCULATED
C      NTAPE     UNIT NUMBER OF FILE FROM WHICH COEFFICIENTS ARE READ
C      IPATT     HEPPNER-MAYNARD PATTERN TYPE
C      NNMAX     NUMBER OF COEFFICIENTS
C      A         RADIUS OF ELLIPSE IN X (SUNWARD) DIRECTION
C      B         RADIUS OF ELLIPSE IN Y (DUSKWARD) DIRECTION
C      DX        OFFSET OF ELLIPSE IN X DIRECTION
C      DY        OFFSET OF ELLIPSE IN Y DIRECTION
```

90/06/30
12:03:44

msml.for

28

```
C      XCO      COEFFICIENTS      MSM17290
C      AHM      HEPPNER-MAYNARD X RADIUS      MSM17300
C      BHM      HEPPNER-MAYNARD Y RADIUS      MSM17310
C      DXHM      HEPPNER-MAYNARD X OFFSET      MSM17320
C      DYHM      HEPPNER-MAYNARD Y OFFSET      MSM17330
C      VMIN      ELECTRIC POTENTIAL MIN FROM HEPPNER-MAYNARD MODEL      MSM17340
C      VMAX      ELECTRIC POTENTIAL MAX FROM HEPPNER-MAYNARD MODEL      MSM17350
C      PCP      MEASURED CROSS-POLAR-CAP POTENTIAL DROP      MSM17360
C      ICNTRL    = 1 MEANS DIFFERENT FORMULAS USED IN REGIONS 1,2,3      MSM17370
C              = 2 H-M USED AT ALL LATITUDES, BUT SCALED TO PCP AND      MSM17380
C              ELLIPSE PARAMETERS      MSM17390
C              =-1 H-M USED AT ALL LATITUDES, UNSCALED      MSM17400
C              =-2 SHOULD NOT BE USED. RESERVED FOR USE WITH RICE      MSM17410
C              CONVECTION MODEL      MSM17420
C              MSM17430
C      OUTPUT:   MSM17440
C      VALUE     ELECTRIC POTENTIAL (VOLTS)      MSM17450
C              MSM17460
C              MSM17470
C              DIMENSION AHM(2,7),BHM(2,7),DXHM(2,7),DYHM(2,7)      MSM17480
C              DIMENSION VMIN(7),VMAX(7)      MSM17490
C              DIMENSION A(3), B(3), DX(3),DY(3)      MSM17500
C              DIMENSION COEF(18,18),P(18,18)      MSM17510
C              MSM17520
C              DIMENSION DP(18,18),CONST(18,18),SP(18),      MSM17530
C              X   CP(18),FN(18),FM(18)      MSM17540
C              DIMENSION XCO(18,18),MLBL(5)      MSM17550
C              MSM17560
C              DATA CMIN,CMAX/50.,90./      MSM17570
C              DATA P,CONST/648*0./      MSM17580
C      80 CONTINUE      MSM17590
C              MSM17600
C              NMAX = NNMAX      MSM17610
C              MSM17620
C              PI = 3.1415927      MSM17630
C              VALUE=-1.E-9      MSM17640
C              IF (TLAT.GT.CMIN)GOTO70      MSM17650
C              PRINT *, 'LATITUDE ',TLAT,' OUT OF MODEL RANGE'      MSM17660
C              RETURN      MSM17670
C              MSM17680
C              MSM17690
C      70 CONTINUE      MSM17700
C              MSM17710
C      SEGMENT FOR SCALING GRID-PT. LOCATION TO FIT H-M ELLIPSE.      MSM17720
C      TCOL = 90.-TLAT      MSM17730
C      TLONG = TLON - PI      MSM17740
C      IF(ICNTRL.GE.0) THEN      MSM17750
C          XX = DXHM(1,IPATT)+AHM(1,IPATT)*(TCOL*COS(TLONG)-DX(1))/A(1)      MSM17760
C          YY = DYHM(1,IPATT)+BHM(1,IPATT)*(TCOL*SIN(TLONG)-DY(1))/B(1)      MSM17770
C      ELSE      MSM17780
C          XX = TCOL*COS(TLONG)      MSM17790
C          YY = TCOL*SIN(TLONG)      MSM17800
C      END IF      MSM17810
C      XCOL = SQRT(XX**2 + YY**2)      MSM17820
C      XLAT = 90.-XCOL      MSM17830
C      XL = ATAN2(YY,XX)      MSM17840
C      XLON = XL+PI      MSM17850
C      ALPHA=2./(CMAX-CMIN)      MSM17860
C      BETA=1.-ALPHA*CMAX      MSM17870
C      CT=XLAT*ALPHA+BETA      MSM17880
C      ST=SQRT(1.-CT*CT)      MSM17890
C      SPH=SIN(XLON)      MSM17900
C      CPH=COS(XLON)      MSM17910
C              MSM17920
```

90/06/30
12:03:44

msm1.for

29

```
      IF (P(1,1).NE.0.) GOTO 3
      P(1,1)=1.0
      DP(1,1)=0.0
      SP(1)=0.0
      CP(1)=1.0
      DO 2 N=2,18
      FN(N)=N
      DO 2 M=1,N
      FM(M)=M-1
2  CONST(N,M)=FLOAT((N-2)**2-(M-1)**2)/
      X  FLOAT((2*N-3)*(2*N-5))
C
      3 SP(2)=SPH
      P(1,1)=1.
      CP(2)=CPH
      DO 4 M=3,NMAX
      SP(M)=SP(2)*CP(M-1)+CP(2)*SP(M-1)
      4 CP(M)=CP(2)*CP(M-1)-SP(2)*SP(M-1)
C
      VALUE=COEF(1,1)
C
      DO 8 N=2,NMAX
      DO 8 M=1,N
C
      IF (N.NE.M) GOTO 6
      P(N,N)=ST*P(N-1,N-1)
C
      DP(N,N)=ST*DP(N-1,N-1)+CT*P(N-1,N-1)
      GOTO 8
      6 IF (N.NE.2) P(N,M)=CT*P(N-1,M)-CONST(N,M)*P(N-2,M)
      IF (N.EQ.2) P(N,M)=CT*P(N-1,M)
C
      DP(N,M)=CT*DP(N-1,M)-ST*P(N-1,M)-
C
      X  CONST(N,M)*DP(N-2,M)
C
C
      8 CONTINUE
C
      P(1,1)=P(1,1)*XCO(1,1)
C
      VALUE=COEF(1,1)*P(1,1)
      DO 10 N=2,NMAX
      DO 10 M=1,N
      IF (M.EQ.1) GOTO 11
      POL=P(N,M)*XCO(N,M)
      P(M-1,N)=CP(M)*POL
      P(N,M)=SP(M)*POL
      VALUE=VALUE+P(M-1,N)*COEF(M-1,N)+
      X  P(N,M)*COEF(N,M)
      GOTO 10
      11 P(N,M)=P(N,M)*XCO(N,M)
      VALUE=VALUE+P(N,M)*COEF(N,M)
      10 CONTINUE
      IF(ICNTRL.NE.-1) THEN
      VALUE = (PCP/(VMAX(IPATT)-VMIN(IPATT)))*VALUE*1000.
      ELSE
      VALUE = 1000.*VALUE
      END IF
C
C
      RETURN
      END
C
C
      SUBROUTINE LOW(COL,ALO,VLOW,DEQDT,AEQEDG,DXEQEG,VBAR,G)
```

MSM17930
MSM17940
MSM17950
MSM17960
MSM17970
MSM17980
MSM17990
MSM18000
MSM18010
MSM18020
MSM18030
MSM18040
MSM18050
MSM18060
MSM18070
MSM18080
MSM18090
MSM18100
MSM18110
MSM18120
MSM18130
MSM18140
MSM18150
MSM18160
MSM18170
MSM18180
MSM18190
MSM18200
MSM18210
MSM18220
MSM18230
MSM18240
MSM18250
MSM18260
MSM18270
MSM18280
MSM18290
MSM18300
MSM18310
MSM18320
MSM18330
MSM18340
MSM18350
MSM18360
MSM18370
MSM18380
MSM18390
MSM18400
MSM18410
MSM18420
MSM18430
MSM18440
MSM18450
MSM18460
MSM18470
MSM18480
MSM18490
MSM18500
MSM18510
MSM18520
MSM18530
MSM18540
MSM18550
MSM18560

90/06/30
12:03:44

30

msml.for

```
C  PURPOSE:  RETURN THE ELECTRIC POTENTIAL AT A SPECIFIC LOCATION IN
C            THE LOW-LATITUDE REGION OF THE ELECTRIC FIELD MODEL.
C
C  REFERENCE:  SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-
C              87-K-0001.
C
C  INPUTS:
C      COL          COLATITUDE (RADIAN)S)
C      ALO          MLT (RADIAN)S)
C      DEQDT        TIME RATE OF CHANGE OF EQUATORWARD EDGE OF AURORAL
C                  ZONE (DEGREE)S/HOUR)
C      AEQEDG       RADIUS OF ELLIPSE 3 IN X (SUNWARD) DIRECTION
C      DXEQEG       OFFSET OF ELLIPSE 3 IN X DIRECTION
C      VBAR         AVERAGE POTENTIAL AT LOW LATITUDES
C      G            WEIGHTING FACTOR
C
C  OUTPUT:
C      VLOW         ELECTRIC POTENTIAL (VOLTS)
C
C
C      PI = 3.1415927
C      AK = 1.38
C      SA = SIN((AEQEDG-DXEQEG)*PI/180.)
C      CF = 2.75*1000.*1.576*G*DEQDT*SA/SIN(.4363)
C      FAC = .6103*SIN(ALO)-.0154*SIN(2.*ALO)-.021*SIN(3.*ALO)
C      FAC = FAC - .1092*COS(ALO)+.1676*COS(2.*ALO)-.0314*COS(3.*ALO)
C      VLOW = CF*FAC*(SA/SIN(COL))**AK + VBAR
C      RETURN
C      END
C
C
C      SUBROUTINE AURORA(COL,ALO,VAZ,VBAZ,TB,TC,J,JMAX,JWRAP)
C
C  PURPOSE:  PROVIDE THE ELECTRIC POTENTIAL AT A SPECIFIC LOCATION IN
C            THE AURORAL REGION OF THE ELECTRIC FIELD MODEL.
C
C  REFERENCE:  SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-
C              87-K-0001.
C
C  INPUTS:
C      COL          COLATITUDE OF POINT
C      ALO          MLT OF POINT
C      VBAZ         AURORAL CONTRIBUTION TO POTENTIAL
C      TB          LOCATION OF BOUNDARY B FOR SPECIFIC LOCAL TIME
C      TC          LOCATION OF BOUNDARY C FOR SPECIFIC LOCAL TIME
C      J           LOCAL TIME INDEX
C      JMAX        MAXIMUM NUMBER OF PTS IN J (LONGITUDE) DIRECTION
C      JWRAP       NUMBER OF PTS OF OVERLAP IN J DIRECTION
C
C  OUTPUT:
C      VAZ         ELECTRIC POTENTIAL (VOLTS)
C
C
C      PARAMETER (JDIM=51)
C      DIMENSION VBAZ(JDIM)
C
C
C      IF (JDIM.NE.JMAX) THEN
```

MSM18570
MSM18580
MSM18590
MSM18600
MSM18610
MSM18620
MSM18630
MSM18640
MSM18650
MSM18660
MSM18670
MSM18680
MSM18690
MSM18700
MSM18710
MSM18720
MSM18730
MSM18740
MSM18750
MSM18760
MSM18770
MSM18780
MSM18790
MSM18800
MSM18810
MSM18820
MSM18830
MSM18840
MSM18850
MSM18860
MSM18870
MSM18880
MSM18890
MSM18900
MSM18910
MSM18920
MSM18930
MSM18940
MSM18950
MSM18960
MSM18970
MSM18980
MSM18990
MSM19000
MSM19010
MSM19020
MSM19030
MSM19040
MSM19050
MSM19060
MSM19070
MSM19080
MSM19090
MSM19100
MSM19110
MSM19120
MSM19130
MSM19140
MSM19150
MSM19160
MSM19170
MSM19180
MSM19190
MSM19200

90/06/30
12:03:44

msm1.for

31

```
      WRITE(6,147)JDIM,JMAX
147  FORMAT(1H,'JDIM = ',I4,' LTDIM = ',I4,' IN AURORA.')
      END IF
      DPWIDE = 1.
      DPAMP = 2.0
      PI = 3.1415927
      ALMIN = PI - .5*DPWIDE
      ALMAX = PI + .5*DPWIDE
C
      IF(ALO.GT.ALMIN.AND.ALO.LT.ALMAX) THEN
          PHIP = COS((PI*(ALO-PI))/DPWIDE)
      ELSE
          PHIP = 0.
      END IF
      IF (J.EQ.1) THEN
          JM1 = JMAX - JWRAP
      ELSE
          JM1 = J-1
      END IF
C
      IF (J.EQ.JMAX) THEN
          JP1 = JWRAP + 1
      ELSE
          JP1 = J+1
      END IF
C
C
      DVBDPH = (VBAZ(JP1) - VBAZ(JM1))*PI/FLOAT(JMAX-3)
      DENOM = TC - COL
      ANUM = TC - TB
      IF (DENOM.LT.0.001) DENOM = 0.001
C
      CORR = -DVBDPH*PHIP*DPAMP*6.75*(TC-COL)**2*(COL-TB)/(TC-TB)**3
C
      R = 1.
      DTHETA = (1.25-.75*COS(ALO-.75*PI))*(TC-TB)
      FAC2 = (1.+(TC-COL)**2/DTHETA**2)**(-R)
      FAC3 = (1.+(TC-TB)**2/DTHETA**2)**(-R)
      VAZ = VBAZ(J)*(1.-FAC2)/(1.-FAC3) + CORR
C
      RETURN
      END
C
C
C
      SUBROUTINE REG1(COL,AL,V1,TA,TB,VAA,VBB,DVAA,DVBB)
C
C  PURPOSE:  PROVIDE ELECTRIC POTENTIAL IN REGION 1 OF THE ELECTRIC
C            FIELD MODEL
C
C
C  REFERENCE:  SECTION 2.4 OF THE FINAL REPORT FOR CONTRACT #F19628-
C              87-K-0001.
C
C  INPUTS:
C      COL          COLATITUDE (RADIAN)
C      AL           MLT
C      TA           LOCATION OF BOUNDARY A
C      TB           LOCATION OF BOUNDARY B
C      VAA          ELECTRIC POTENTIAL FOR PT ON BOUNDARY A
C      VBB          ELECTRIC POTENTIAL FOR PT ON BOUNDARY B
C      DVAA         DERIVATIVE OF V FOR PT ON BOUNDARY A
C      DVBB         DERIVATIVE OF V FOR PT ON BOUNDARY B
```

MSM19210
MSM19220
MSM19230
MSM19240
MSM19250
MSM19260
MSM19270
MSM19280
MSM19290
MSM19300
MSM19310
MSM19320
MSM19330
MSM19340
MSM19350
MSM19360
MSM19370
MSM19380
MSM19390
MSM19400
MSM19410
MSM19420
MSM19430
MSM19440
MSM19450
MSM19460
MSM19470
MSM19480
MSM19490
MSM19500
MSM19510
MSM19520
MSM19530
MSM19540
MSM19550
MSM19560
MSM19570
MSM19580
MSM19590
MSM19600
MSM19610
MSM19620
MSM19630
MSM19640
MSM19650
MSM19660
MSM19670
MSM19680
MSM19690
MSM19700
MSM19710
MSM19720
MSM19730
MSM19740
MSM19750
MSM19760
MSM19770
MSM19780
MSM19790
MSM19800
MSM19810
MSM19820
MSM19830
MSM19840

90/06/30
12:03:44

msm1.for

32

OUTPUT:

V1 ELECTRIC POTENTIAL FOR A POINT IN REGION 1 (VOLTS)

DT = TB - TA

V1 = VAA + DVAA*(COL-TA)

V1 = V1 + (3.*(VBB-VAA) - (2.*DVAA+DVBB)*DT)*(COL-TA)**2/DT**2

V1 = V1 + (2.*(VAA-VBB) + (DVAA+DVBB)*DT)*(COL-TA)**3/DT**3

RETURN

END

FUNCTION THET(AA,BB,XC,YC,PHI)

THIS FUNCTION GIVES SOLUTION TO EQUATION FOR ELLIPSE IN
IN FLAT POLAR COORDINATES. IT SPECIFIES COLATITUDE AS A
FUNCTION OF LOCAL TIME.

INPUTS:

AA RADIUS OF ELLIPSE IN X (SUNWARD) DIRECTION
BB RADIUS OF ELLIPSE IN Y (DUSKWARD) DIRECTION
XC X DISPLACEMENT OF ELLIPSE
YC Y DISPLACEMENT OF ELLIPSE
PHI MLT HOUR ANGLE

OUTPUT:

THET COLATITUDE OF INPUT ELLIPSE AT GIVEN PHI

CPHI = COS(PHI)

SPHI = SIN(PHI)

CA = (CPHI/AA)**2 + (SPHI/BB)**2

CB = -XC*CPHI/AA**2 - YC*SPHI/BB**2

CC = (XC/AA)**2 + (YC/BB)**2 - 1.

THET = (-CB+SQRT(CB**2-CA*CC))/CA

RETURN

END

SUBROUTINE INPUT(NTAPE,COEF,IPATT,XCO,AHM,BHM,DXHM,
1 DYHM,VMIN, VMAX)

PURPOSE: READ IN NECESSARY INPUT FOR THE COMPUTATION OF THE
HEPPNER-MAYNARD ELECTRIC FIELD MODEL

DIMENSION COEF(18,18)

DIMENSION DXHM(2,7),DYHM(2,7)

DIMENSION XCO(18,18), AHM(2,7),BHM(2,7)

DIMENSION VMIN(7),VMAX(7),YCO(18,18)

DIMENSION AAHM(2,7),BBHM(2,7),DDXHM(2,7)

DIMENSION DDYHM(2,7), VVMAX(7), VVMIN(7)

CHARACTER*50 MLBL

DATA (YCO(1,J),J=1,18)/

X .282095E+00, .488603E+00, .109255E+01, .228523E+01,

X .468333E+01, .951188E+01, .192265E+02, .387523E+02,

X .779645E+02, .156658E+03, .314501E+03, .630964E+03,

X .126523E+04, .253611E+04, .508196E+04, .101809E+05,

X .203918E+05, .408366E+05/

DATA (YCO(2,J),J=1,18)/

X .488603E+00, .488603E+00, .546274E+00, .144531E+01,

X .331161E+01, .719031E+01, .151999E+02, .316411E+02,

X .652298E+02, .133599E+03, .272366E+03, .553392E+03,

MSM19850

MSM19860

MSM19870

MSM19880

MSM19890

MSM19900

MSM19910

MSM19920

MSM19930

MSM19940

MSM19950

MSM19960

MSM19970

MSM19980

MSM19990

MSM20000

MSM20010

MSM20020

MSM20030

MSM20040

MSM20050

MSM20060

MSM20070

MSM20080

MSM20090

MSM20100

MSM20110

MSM20120

MSM20130

MSM20140

MSM20150

MSM20160

MSM20170

MSM20180

MSM20190

MSM20200

MSM20210

MSM20220

MSM20230

MSM20240

MSM20250

MSM20260

MSM20270

MSM20280

MSM20290

MSM20300

MSM20310

MSM20320

MSM20330

MSM20340

MSM20350

MSM20360

MSM20370

MSM20380

MSM20390

MSM20400

MSM20410

MSM20420

MSM20430

MSM20440

MSM20450

MSM20460

MSM20470

MSM20480

msm1.for

```
X .112151E+04, .226837E+04, .458082E+04, .923904E+04,
X .186151E+05, .374743E+05/
DATA (YCO( 3,J),J=1,18)/
X .946175E+00, .109255E+01, .546274E+00, .590044E+00,
X .177013E+01, .440314E+01, .101333E+02, .223736E+02,
X .481754E+02, .102038E+03, .213661E+03, .443701E+03,
X .915709E+03, .188083E+04, .384856E+04, .785168E+04,
X .159791E+05, .324537E+05/
DATA (YCO( 4,J),J=1,18)/
X .186588E+01, .228523E+01, .144531E+01, .590044E+00,
X .625836E+00, .207566E+01, .555021E+01, .134918E+02,
X .310971E+02, .693209E+02, .151081E+03, .324033E+03,
X .686782E+03, .144253E+04, .300864E+04, .623988E+04,
X .128827E+05, .264983E+05/
DATA (YCO( 5,J),J=1,18)/
X .370249E+01, .468333E+01, .331161E+01, .177013E+01,
X .625836E+00, .656382E+00, .236662E+01, .674590E+01,
X .172496E+02, .414272E+02, .955522E+02, .214328E+03,
X .471128E+03, .102002E+04, .218269E+04, .462762E+04,
X .973844E+04, .203694E+05/
DATA (YCO( 6,J),J=1,18)/
X .736787E+01, .951188E+01, .719031E+01, .440314E+01,
X .207566E+01, .656382E+00, .683184E+00, .264596E+01,
X .798499E+01, .213929E+02, .534153E+02, .127330E+03,
X .293800E+03, .661878E+03, .146420E+04, .319336E+04,
X .688612E+04, .147131E+05/
DATA (YCO( 7,J),J=1,18)/
X .146845E+02, .192265E+02, .151999E+02, .101333E+02,
X .555021E+01, .236662E+01, .683184E+00, .707163E+00,
X .291571E+01, .926339E+01, .259102E+02, .671087E+02,
X .165101E+03, .391572E+03, .903721E+03, .204248E+04,
X .454057E+04, .996084E+04/
DATA (YCO( 8,J),J=1,18)/
X .292940E+02, .387523E+02, .316411E+02, .223736E+02,
X .134918E+02, .674590E+01, .264596E+01, .707163E+00,
X .728927E+00, .317732E+01, .105778E+02, .307916E+02,
X .825507E+02, .209304E+03, .509767E+03, .120459E+04,
X .278052E+04, .629979E+04/
DATA (YCO( 9,J),J=1,18)/
X .584734E+02, .779645E+02, .652298E+02, .481754E+02,
X .300971E+02, .172496E+02, .798499E+01, .291571E+01,
X .728927E+00, .748901E+00, .343190E+01, .119255E+02,
X .360281E+02, .997819E+02, .260366E+03, .650553E+03,
X .157290E+04, .370647E+04/
DATA (YCO(10,J),J=1,18)/
X .116766E+03, .156658E+03, .133599E+03, .102038E+03,
X .693209E+02, .414272E+02, .213929E+02, .926339E+01,
X .317732E+01, .748901E+00, .767395E+00, .368030E+01,
X .133043E+02, .416119E+02, .118840E+03, .318704E+03,
X .816138E+03, .201755E+04/
DATA (YCO(11,J),J=1,18)/
X .233240E+03, .314501E+03, .272366E+03, .213661E+03,
X .151081E+03, .955522E+02, .534153E+02, .259102E+02,
X .105778E+02, .343190E+01, .767395E+00, .784642E+00,
X .392321E+01, .147120E+02, .475361E+02, .139761E+03,
X .384731E+03, .100877E+04/
DATA (YCO(12,J),J=1,18)/
X .465998E+03, .630964E+03, .553392E+03, .443701E+03,
X .324033E+03, .214328E+03, .127330E+03, .671087E+02,
X .307916E+02, .119255E+02, .368030E+01, .784642E+00,
X .800822E+00, .416119E+01, .161472E+02, .537941E+02,
X .162579E+03, .458849E+03/
DATA (YCO(13,J),J=1,18)/
X .931187E+03, .126523E+04, .112151E+04, .915709E+03,
```

```
MSM20490
MSM20500
MSM20510
MSM20520
MSM20530
MSM20540
MSM20550
MSM20560
MSM20570
MSM20580
MSM20590
MSM20600
MSM20610
MSM20620
MSM20630
MSM20640
MSM20650
MSM20660
MSM20670
MSM20680
MSM20690
MSM20700
MSM20710
MSM20720
MSM20730
MSM20740
MSM20750
MSM20760
MSM20770
MSM20780
MSM20790
MSM20800
MSM20810
MSM20820
MSM20830
MSM20840
MSM20850
MSM20860
MSM20870
MSM20880
MSM20890
MSM20900
MSM20910
MSM20920
MSM20930
MSM20940
MSM20950
MSM20960
MSM20970
MSM20980
MSM20990
MSM21000
MSM21010
MSM21020
MSM21030
MSM21040
MSM21050
MSM21060
MSM21070
MSM21080
MSM21090
MSM21100
MSM21110
MSM21120
```

90/06/30
12:03:44

msm1.for

34

```
X .686782E+03, .471128E+03, .293800E+03, .165101E+03,  
X .825507E+02, .360281E+02, .133043E+02, .392321E+01,  
X .800822E+00, .816077E+00, .439471E+01, .176082E+02,  
X .603802E+02, .187325E+03/  
DATA (YCO(14,J),J=1,18) /  
X .186100E+04, .253611E+04, .226837E+04, .188083E+04,  
X .144253E+04, .102002E+04, .661878E+03, .391572E+03,  
X .209304E+03, .997819E+02, .416119E+02, .147120E+02,  
X .416119E+01, .816077E+00, .830522E+00, .462415E+01,  
X .190939E+02, .672889E+02/  
DATA (YCO(15,J),J=1,18) /  
X .371962E+04, .508196E+04, .458082E+04, .384866E+04,  
X .300864E+04, .218269E+04, .146420E+04, .903721E+03,  
X .509767E+03, .260366E+03, .118840E+03, .475361E+02,  
X .161472E+02, .439471E+01, .830522E+00, .844251E+00,  
X .484985E+01, .206029E+02/  
DATA (YCO(16,J),J=1,18) /  
X .743510E+04, .101809E+05, .923904E+04, .785168E+04,  
X .623988E+04, .462762E+04, .319336E+04, .204248E+04,  
X .120459E+04, .650553E+03, .318704E+03, .139761E+03,  
X .537941E+02, .176082E+02, .462415E+01, .844251E+00,  
X .857341E+00, .507210E+01/  
DATA (YCO(17,J),J=1,18) /  
X .148629E+05, .203918E+05, .186151E+05, .159791E+05,  
X .128827E+05, .973844E+04, .688612E+04, .454057E+04,  
X .278052E+04, .157290E+04, .816138E+03, .384731E+03,  
X .162579E+03, .603802E+02, .190939E+02, .484985E+01,  
X .857341E+00, .869857E+00/  
DATA (YCO(18,J),J=1,18) /  
X .297130E+05, .408366E+05, .374743E+05, .324537E+05,  
X .264983E+05, .203694E+05, .147131E+05, .996084E+04,  
X .629979E+04, .370647E+04, .201755E+04, .100877E+04,  
X .458849E+03, .187325E+03, .672889E+02, .206029E+02,  
X .507210E+01, .869857E+00/  
C  
DATA (AAHM(1,IP),IP=1,7)/17.45,12.13,16.06,13.72,14.79,12.82,15./  
DATA (AAHM(2,IP),IP=1,7)/20.,16.17,18.88,19.31,18.3,16.6,17.93/  
DATA (BBHM(1,IP),IP=1,7)/14.26,13.78,14.31,13.78,14.73,11.70,12.07/  
DATA (BBHM(2,IP),IP=1,7)/16.97,16.7,17.07,17.23,17.93,17.5,16.6/  
DATA (DDXHM(1,IP),IP=1,7)/-2.66,-5.53,-3.19,-.11,-2.45,-1.65,-2.23/  
DATA (DDXHM(2,IP),IP=1,7)/-3.3,-7.45,-4.1,-1.76,-2.77,-2.98,-3.14/  
DATA (DDYHM(1,IP),IP=1,7)/1.6,.8,1.12,.05,.48,3.4,1.22/  
DATA (DDYHM(2,IP),IP=1,7)/1.33,.32,1.54,.85,.69,1.86,.21/  
DATA (VVMAX(IP),IP=1,7)/34007.,55354.,14390.,11287.,9329.,13249.,  
1 13221./  
DATA (VVMIN(IP),IP=1,7)/-42280.,-16003.,-60935.,-16250.,-12947.,  
1 -14428.,-13460./  
C  
C  
DO 4 IP = 1,7  
DO 3 N=1,2  
AAHM(N,IP) = AAHM(N,IP)  
BBHM(N,IP) = BBHM(N,IP)  
DXHM(N,IP) = DDXHM(N,IP)  
DYHM(N,IP) = DDYHM(N,IP)  
3 CONTINUE  
VMAX(IP) = VVMAX(IP)  
VMIN(IP) = VVMIN(IP)  
4 CONTINUE  
C  
C  
C INITIALIZE COEF FROM UNIT NTAPE  
C  
C  
DO 1 I=1,18
```

MSM21130
MSM21140
MSM21150
MSM21160
MSM21170
MSM21180
MSM21190
MSM21200
MSM21210
MSM21220
MSM21230
MSM21240
MSM21250
MSM21260
MSM21270
MSM21280
MSM21290
MSM21300
MSM21310
MSM21320
MSM21330
MSM21340
MSM21350
MSM21360
MSM21370
MSM21380
MSM21390
MSM21400
MSM21410
MSM21420
MSM21430
MSM21440
MSM21450
MSM21460
MSM21470
MSM21480
MSM21490
MSM21500
MSM21510
MSM21520
MSM21530
MSM21540
MSM21550
MSM21560
MSM21570
MSM21580
MSM21590
MSM21600
MSM21610
MSM21620
MSM21630
MSM21640
MSM21650
MSM21660
MSM21670
MSM21680
MSM21690
MSM21700
MSM21710
MSM21720
MSM21730
MSM21740
MSM21750
MSM21760

90/06/30
12:03:44

msml.for

35

```
DO 2 J=1,18
  COEF(I,J) = 0.
  XCO(I,J) = YCO(I,J)
2  CONTINUE
1  CONTINUE
C
C  OPEN FILE TO GET ELECTRIC FIELD COEFFICIENTS
C
  OPEN(UNIT=NTAPE,FILE='EFCOEF',STATUS='OLD')
C
  DO 81 IA=1,IPATT
    READ(NTAPE,400,ERR=1001,END=1001)MLBL
1001 CONTINUE
400  FORMAT(A)
    82 READ(NTAPE,*)NNN,MMM,I,J,CCF
C    WRITE(8,100) IA,NNN,MMM,I,J,CCF
C100  FORMAT(1H,'IA,NNN,MMM,I,J,CCF = ',5I4,E15.5)
    IF(NNN.EQ.-1) GO TO 81
    IF(I.GT.18.OR.J.GT.18) THEN
      WRITE(6,737) IA,NNN,MMM,I,J
737  FORMAT(1H,'OUT OF BOUNDS IN INPUT.IA,NNN,MMM,I,J = ',5I5)
    END IF
    COEF(I,J)=CCF
    GO TO 82
81  CONTINUE
C  WRITE(8,401)MLBL
C401  FORMAT(1H,'MLBL = ',5A10)
      NMAX=NNMAX
C
      CLOSE(NTAPE)
C
      RETURN
C
      END
C
C  SUBROUTINE BTRACE(LATDIM,LTDIM,ITMDIM,L,FSTOFF,FEQEDG,FDST,
2      FCLPSE,FTILT,VM,BMIN,XMIN,YMIN,ZMIN,R,P,ALOCT,TETA)
C
C  VERSION 2.02
C
C  DATE: 09.16.89
C
C  PURPOSE: SUBROUTINE TO CALCULATE B FIELD MODELS APPROPRIATE FOR
C            GEOPHYSICAL CONDITIONS AS SPECIFIED IN BFPAR. BTRACE
C            CALCULATES VM, BMIN, XMIN, YMIN, AND ZMIN ON THE MSM SPATIAL
C            GRID BY INTERPOLATING BETWEEN PRECOMPUTED MAGNETIC FIELD
C            MODELS AS PARAMETERIZED BY THE VARIABLES OF BFPAR.
C
C  INPUT:
C      LATDIM  NUMBER OF LATITUDINAL GRID POINTS
C      LTDIM   NUMBER OF LOCAL TIME GRID POINTS (INCL WRAPAROUND)
C      ITMDIM  MAX NUMBER OF TEMPORAL GRID POINTS
C      L       CURRENT TEMPORAL GRID INDEX
C      .....  GEOPHYSICAL PARAMETERS THAT DETERMINE WHICH BFIELD
C              MODELS TO USE
C      FSTOFF  STANDOFF DIST = AUGPAR(STAND)
C      FEQEDG  EQUATORWARD EDGE OF AURORA = AUGPAR(EQEDGE)
C      FDST    DST = AUGPAR(DST)
C      FCLPSE  COLLAPSE PARAMETER = AUGPAR(CLAPSE)
C      FTILT   TILT PARAMETER = AUGPAR(TILTW)
C      ALOCT   LOCAL TIME HOUR ANGLE (MEASURED EAST FROM NOON)
C      TETA    COLATITUDE OF I GRID LINES (RADIAN)
C
C  OUTPUT:
C      VM      (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3
```

MSM21770
MSM21780
MSM21790
MSM21800
MSM21810
MSM21820
MSM21830
MSM21840
MSM21850
MSM21860
MSM21870
MSM21880
MSM21890
MSM21900
MSM21910
MSM21920
MSM21930
MSM21940
MSM21950
MSM21960
MSM21970
MSM21980
MSM21990
MSM22000
MSM22010
MSM22020
MSM22030
MSM22040
MSM22050
MSM22060
MSM22070
MSM22080
MSM22090
MSM22100
MSM22110
MSM22120
MSM22130
MSM22140
MSM22150
MSM22160
MSM22170
MSM22180
MSM22190
MSM22200
MSM22210
MSM22220
MSM22230
MSM22240
MSM22250
MSM22260
MSM22270
MSM22280
MSM22290
MSM22300
MSM22310
MSM22320
MSM22330
MSM22340
MSM22350
MSM22360
MSM22370
MSM22380
MSM22390
MSM22400

90/06/30
12:03:44

msm1.for

36

C	BMIN	B VALUE AT MINIMUM VALUE ALONG FIELD LINE (NT)	MSM22410
C	XMIN	X (GSM) VALUE AT WHICH BMIN OCCURS	MSM22420
C	YMIN	Y (GSM) VALUE AT WHICH BMIN OCCURS	MSM22430
C	ZMIN	Z (GSM) VALUE AT WHICH BMIN OCCURS	MSM22440
C	R	R SQRT(X**2+Y**2+Z**2) AT WHICH BMIN OCCURS	MSM22450
C	P	HOOR ANGLE (RADIAN) MEASURED EASTWARD FROM NOON	MSM22460
C			MSM22470
C	WORK:		MSM22480
C	WORK	WORK ARRAY (NEEDED FOR BFIELD RETRIEVAL AND	MSM22490
C		INTERPOLATION)	MSM22500
C			MSM22510
C	REFERENCE:		MSM22520
C		FIVE DIMENSIONAL INTERPOLATION IS BASED ON MULTILINEAR	MSM22530
C		INTERPOLATION SCHEME GIVEN BY PRESS ET AL. IN	MSM22540
C		NUMERICAL RECIPES, CAMBRIDGE UNIVERSITY PRESS, 1987, PP 95-97	MSM22550
C			MSM22560
C			MSM22570
C	PROGRAMMER:	R.W. SPIRO	MSM22580
C			MSM22590
C			MSM22600
C	PARAMETER(IIDIM=62,JJDIM=51)		MSM22610
C	COMMON /GRID/DLAM,DPSI,RI,JWRAP		MSM22620
C			MSM22630
C	LOGICAL*1 BFEXST(2,2,2,2,2)		MSM22640
C			MSM22650
C			MSM22660
C	DIMENSION WORK(IIDIM,JJDIM,2,2,2,2,2)		MSM22670
C	DIMENSION VM(LATDIM,LTDIM,ITMDIM)		MSM22680
C	DIMENSION BMIN(LATDIM,LTDIM),XMIN(LATDIM,LTDIM)		MSM22690
C	DIMENSION YMIN(LATDIM,LTDIM),ZMIN(LATDIM,LTDIM)		MSM22700
C	DIMENSION ALOCT(LATDIM,LTDIM),R(LATDIM,LTDIM,ITMDIM)		MSM22710
C	DIMENSION P(LATDIM,LTDIM,ITMDIM)		MSM22720
C	DIMENSION BFLIM(2,5),BFPAR(5)		MSM22730
C	DIMENSION T(2,5),N(5)		MSM22740
C	DIMENSION SUMC(IIDIM,JJDIM)		MSM22750
C	DIMENSION TETA(LATDIM)		MSM22760
C			MSM22770
C	DATA BFBUSH /1.E20/		MSM22780
C			MSM22790
C			MSM22800
C	CHECK DIMENSIONS		MSM22810
C	IF(IIDIM.NE.LATDIM.OR.JJDIM.NE.LTDIM) THEN		MSM22820
C	WRITE(6,*) 'DIMENSION ERROR IN BTRACE'		MSM22830
C	WRITE(6,*) 'IIDIM= ',IIDIM,' SHOULD EQUAL LATDIM= ',LATDIM		MSM22840
C	WRITE(6,*) 'JJDIM= ',JJDIM,' SHOULD EQUAL LTDIM= ',LTDIM		MSM22850
C	WRITE(6,*) 'STOPPING PGM IN BTRACE; CHECK DIMENSIONS'		MSM22860
C	STOP		MSM22870
C	END IF		MSM22880
C			MSM22890
C			MSM22900
C			MSM22910
C	PI=ATAN2(0.,-1.)		MSM22920
C			MSM22930
C	SET UP BFPAR VECTOR FOR CALLING GETMAT		MSM22940
C	BFPAR(1)=FSTOFF		MSM22950
C	BFPAR(2)=FTILT		MSM22960
C	BFPAR(3)=FEQEDG		MSM22970
C	BFPAR(4)=FDST		MSM22980
C	BFPAR(5)=FCLPSE		MSM22990
C			MSM23000
C			MSM23010
C			MSM23020
C	INITIALIZEBFIELD ARRAYS		MSM23030
C			MSM23040

90/06/30
12:03:44

msm1.for

37

```
DO 15 J=1,LTDIM
DO 16 I=1,LATDIM
  VM(I,J,L)=0.
  BMIN(I,J)=0.
  XMIN(I,J)=0.
  YMIN(I,J)=0.
  ZMIN(I,J)=0.
  SUMC(I,J)=0.
16 CONTINUE
15 CONTINUE
C
  WRITE (6,*) 'GETMAT IS REQUESTING THE FOLLOWING B-MATRICES:'
C
C LOOP OVER FIVE FLAVORS OF B FIELD ARRAYS
  DO 20 NN=1,5
    IWANT=NN
C
C CALL GETMAT TO READ APPROPRIATE ARRAYS FOR INTERPOLATION
C
    CALL GETMAT(IWANT,LATDIM,LTDIM,BFPAR,WORK,BFLIM,BFEXST)
    CALL RMVBSH(WORK,LATDIM,LTDIM)
C
C SET UP WEIGHTING PARAMETERS FOR INTERPOLATION
C
    DO 10 II=1,5
      T(2,II)=(BFPAR(II)-BFLIM(1,II))/(BFLIM(2,II)-BFLIM(1,II))
      T(1,II)=1.-T(2,II)
10 CONTINUE
C
    IF(IWANT.EQ.1) THEN
      WRITE(6,*) 'T ARRAY',T
    END IF
C
C MAIN INTERPOLATION LOOPS
C
  DO 21 N1=1,2
    N(1)=N1
  DO 22 N2=1,2
    N(2)=N2
  DO 23 N3=1,2
    N(3)=N3
  DO 24 N4=1,2
    N(4)=N4
  DO 25 N5=1,2
    N(5)=N5
C
    COEFF=1.
C
    DO 30 M=1,5
      COEFF=COEFF*T(N(M),M)
30 CONTINUE
C
C CHECK TO SEE IF B FIELD MODEL EXISTS
C
C
C    WRITE(6,*) 'N1,N2,N3,N4,N5',N1,N2,N3,N4,N5
C    WRITE(6,*) 'COEFF',COEFF
C    WRITE(6,*) 'BFEXST',BFEXST(N1,N2,N3,N4,N5)
C    IF(.NOT.BFEXST(N1,N2,N3,N4,N5)) THEN
      COEFF=0.
    END IF
C
C
C
```

MSM23050
MSM23060
MSM23070
MSM23080
MSM23090
MSM23100
MSM23110
MSM23120
MSM23130
MSM23140
MSM23150
MSM23160
MSM23170
MSM23180
MSM23190
MSM23200
MSM23210
MSM23220
MSM23230
MSM23240
MSM23250
MSM23260
MSM23270
MSM23280
MSM23290
MSM23300
MSM23310
MSM23320
MSM23330
MSM23340
MSM23350
MSM23360
MSM23370
MSM23380
MSM23390
MSM23400
MSM23410
MSM23420
MSM23430
MSM23440
MSM23450
MSM23460
MSM23470
MSM23480
MSM23490
MSM23500
MSM23510
MSM23520
MSM23530
MSM23540
MSM23550
MSM23560
MSM23570
MSM23580
MSM23590
MSM23600
MSM23610
MSM23620
MSM23630
MSM23640
MSM23650
MSM23660
MSM23670
MSM23680

C	LOOP OVER SPATIAL GRID	MSM23690
C		MSM23700
	IF(IWANT.EQ.1) THEN	MSM23710
	DO 40 J=1,LTDIM	MSM23720
	DO 50 I=1,LATDIM	MSM23730
	IF (WORK(I,J,N1,N2,N3,N4,N5) .LT.BFBUSH) THEN	MSM23740
	XMIN(I,J)=XMIN(I,J)+COEFF*	MSM23750
2	WORK(I,J,N1,N2,N3,N4,N5)	MSM23760
	ENDIF	MSM23770
50	CONTINUE	MSM23780
40	CONTINUE	MSM23790
C	WRITE(6,*)' XMIN VALUES'	MSM23800
C	WRITE(6,*)(XMIN(I,33),I=20,26)	MSM23810
C	WRITE(6,*)(XMIN(I,12),I=20,26)	MSM23820
C	WRITE(6,*)' WORK VALUES '	MSM23830
C	WRITE(6,*)(WORK(I,33,N1,N2,N3,N4,N5),I=20,26)	MSM23840
C	WRITE(6,*)(WORK(I,12,N1,N2,N3,N4,N5),I=20,26)	MSM23850
C	WRITE(6,*)N1,N2,N3,N4,N5	MSM23860
C		MSM23870
	ELSE IF(IWANT.EQ.2) THEN	MSM23880
	DO 60 J=1,LTDIM	MSM23890
	DO 70 I=1,LATDIM	MSM23900
	IF (WORK(I,J,N1,N2,N3,N4,N5) .LT.BFBUSH) THEN	MSM23910
	YMIN(I,J)=YMIN(I,J)+COEFF*	MSM23920
2	WORK(I,J,N1,N2,N3,N4,N5)	MSM23930
	END IF	MSM23940
70	CONTINUE	MSM23950
60	CONTINUE	MSM23960
C		MSM23970
	ELSE IF(IWANT.EQ.3) THEN	MSM23980
	DO 80 J=1,LTDIM	MSM23990
	DO 90 I=1,LATDIM	MSM24000
	IF (WORK(I,J,N1,N2,N3,N4,N5) .LT.BFBUSH) THEN	MSM24010
	ZMIN(I,J)=ZMIN(I,J)+COEFF*	MSM24020
2	WORK(I,J,N1,N2,N3,N4,N5)	MSM24030
	END IF	MSM24040
90	CONTINUE	MSM24050
80	CONTINUE	MSM24060
C		MSM24070
	ELSE IF(IWANT.EQ.4) THEN	MSM24080
	DO 100 J=1,LTDIM	MSM24090
	DO 110 I=1,LATDIM	MSM24100
	IF (WORK(I,J,N1,N2,N3,N4,N5) .LT.BFBUSH) THEN	MSM24110
	BMIN(I,J)=BMIN(I,J)+COEFF*	MSM24120
2	WORK(I,J,N1,N2,N3,N4,N5)	MSM24130
	END IF	MSM24140
110	CONTINUE	MSM24150
100	CONTINUE	MSM24160
C		MSM24170
	ELSE	MSM24180
	DO 120 J=1,LTDIM	MSM24190
	DO 130 I=1,LATDIM	MSM24200
	IF (WORK(I,J,N1,N2,N3,N4,N5) .LT.BFBUSH) THEN	MSM24210
	VM(I,J,L)=VM(I,J,L)+COEFF*	MSM24220
2	WORK(I,J,N1,N2,N3,N4,N5)	MSM24230
	SUMC(I,J)=SUMC(I,J)+COEFF	MSM24240
	END IF	MSM24250
130	CONTINUE	MSM24260
120	CONTINUE	MSM24270
	ENDIF	MSM24280
25	CONTINUE	MSM24290
24	CONTINUE	MSM24300
23	CONTINUE	MSM24310
22	CONTINUE	MSM24320

90/06/30
12:03:44

msml.for

39

```
21      CONTINUE
20      CONTINUE
C
C      CALL OUTP(SUMC,IIDIM,JJDIM,10,IIDIM,1,3,JJDIM-3,1,1., 'SUMC',6,132)
C
C      CORRECT FOR MISSING B FIELD MATRICES
C
      DO 140 J=1,LTDIM
      DO 150 I=1,LATDIM
      IF(SUMC(I,J).GT.1.E-10) THEN
      VM(I,J,L)=VM(I,J,L)/SUMC(I,J)
      BMIN(I,J)=BMIN(I,J)/SUMC(I,J)
      XMIN(I,J)=XMIN(I,J)/SUMC(I,J)
      YMIN(I,J)=YMIN(I,J)/SUMC(I,J)
      ZMIN(I,J)=ZMIN(I,J)/SUMC(I,J)
      END IF
150      CONTINUE
140      CONTINUE
C
C
C
C
C      ROTATE BFIELD ARRAYS TO MATCH GRID ORIENTATION
C      NOTE: IN THE CALLS TO BFGYRO, R IS BEING USED AS A WORK ARRAY
C
      CALL BFGYRO(LATDIM,LTDIM,VM(1,1,L),R(1,1,L),ALOCT)
      CALL BFGYRO(LATDIM,LTDIM,BMIN,R(1,1,L),ALOCT)
      CALL BFGYRO(LATDIM,LTDIM,XMIN,R(1,1,L),ALOCT)
      CALL BFGYRO(LATDIM,LTDIM,YMIN,R(1,1,L),ALOCT)
      CALL BFGYRO(LATDIM,LTDIM,ZMIN,R(1,1,L),ALOCT)
C
C
      DO 160 J=1,LTDIM
      DO 170 I=1,LATDIM
      R(I,J,L)=SQRT(XMIN(I,J)**2+YMIN(I,J)**2+ZMIN(I,J)**2)
C
      IF(XMIN(I,J).EQ.0..AND.YMIN(I,J).EQ.0.) THEN
      P(I,J,L)=0.
      ELSE
      P(I,J,L)=ATAN2(YMIN(I,J),XMIN(I,J))
      IF(P(I,J,L).LT.0.) THEN
      P(I,J,L)=P(I,J,L)+2.*PI
      END IF
      END IF
C
170      CONTINUE
160      CONTINUE
C
      RETURN
      END
      SUBROUTINE BFGYRO(LATDIM,LTDIM,ARRAY,WORK,ALOCT)
C
C      VERSION 1.00
C
C      DATE: 08.29.89
C
C      PURPOSE: SUBROUTINE TO ROTATE FIXED GRID B FIELD ARRAYS TO MATCH
C      ROTATING COORDINATE SYSTEM
C
C      NOTE: THIS SUBROUTINE USES ARRAY AS BOTH AN INPUT AND OUTPUT
C      ARRAY. ON INPUT, ARRAY CONTAINS THE UNROTATED DATA; ON
C      OUTPUT, IT CONTAINS THE APPROPRIATELY ROTATED DATA.
C
C      INPUTS:
C      LATDIM      NUMBER OF LATITUDINAL GRID LINES
C      LTDIM       NUMBER OF LOCAL TIME (LONGITUDE) GRID LINES
```

MSM24330
MSM24340
MSM24350
MSM24360
MSM24370
MSM24380
MSM24390
MSM24400
MSM24410
MSM24420
MSM24430
MSM24440
MSM24450
MSM24460
MSM24470
MSM24480
MSM24490
MSM24500
MSM24510
MSM24520
MSM24530
MSM24540
MSM24550
MSM24560
MSM24570
MSM24580
MSM24590
MSM24600
MSM24610
MSM24620
MSM24630
MSM24640
MSM24650
MSM24660
MSM24670
MSM24680
MSM24690
MSM24700
MSM24710
MSM24720
MSM24730
MSM24740
MSM24750
MSM24760
MSM24770
MSM24780
MSM24790
MSM24800
MSM24810
MSM24820
MSM24830
MSM24840
MSM24850
MSM24860
MSM24870
MSM24880
MSM24890
MSM24900
MSM24910
MSM24920
MSM24930
MSM24940
MSM24950
MSM24960

90/06/30
12:03:44

msm1.for

40

```
C      ARRAY          UNROTATED MAGNETIC FIELD ARRAY          MSM24970
C      WORK           INTERNAL WORKING ARRAY                  MSM24980
C      ALOCT          LOCAL TIME HOUR ANGLE (MEASURED EAST FROM NOON) MSM24990
C                                     (RADIAN)                  MSM25000
C                                     (RADIAN)                  MSM25010
C      OUTPUT:        MSM25020
C      ARRAY          ROTATED MAGNETIC FIELD ARRAY          MSM25030
C                                     MSM25040
C                                     MSM25050
C      COMMON /GRID/DLAM,DPSI,RI,JWRAP          MSM25060
C                                     MSM25070
C      DIMENSION ARRAY (LATDIM,LTDIM),WORK (LATDIM,LTDIM)    MSM25080
C      DIMENSION ALOCT (LATDIM,LTDIM)             MSM25090
C      PI=ATAN2(0.,-1.)          MSM25100
C      CALCULATE ROTATION OF COORDINATE SYSTEM          MSM25110
C      ROT=ALOC(1,3)*(LTDIM-JWRAP)/(2.*PI)          MSM25120
C      JROT=ROT          MSM25130
C      F=ROT-FLOAT(JROT)          MSM25140
C      DO 10 J=1,LTDIM          MSM25150
C      J1=J+JROT          MSM25160
C      J1=MOD(J1,LTDIM-JWRAP)          MSM25170
C      IF(J1.EQ.0) J1=LTDIM-JWRAP          MSM25180
C      J2=J1+1          MSM25190
C      DO 20 I=1,LATDIM          MSM25200
C      WORK(I,J)=(1.-F)*ARRAY(I,J1)+F*ARRAY(I,J2)          MSM25210
C      20 CONTINUE          MSM25220
C      10 CONTINUE          MSM25230
C      PUT ROTATED ARRAY BACK INTO ARRAY          MSM25240
C      DO 30 J=1,LTDIM          MSM25250
C      DO 40 I=1,LATDIM          MSM25260
C      ARRAY(I,J)=WORK(I,J)          MSM25270
C      40 CONTINUE          MSM25280
C      30 CONTINUE          MSM25290
C      RETURN          MSM25300
C      END          MSM25310
C      SUBROUTINE PPTM(TSTART,TSTOP,DTNOM,SCTFRC,IFLAV,BNDMAX, MSM25320
1      ALPHA,BETA,BIR,ITMMAX,IDIM,JDIM,IEDIM,ITMDIM,IRDIM, MSM25330
2      NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT, MSM25340
3      THRMAT,TIMTAG,ALAM,ALMDEL,VM,R,P,VMLOSS,V,ETABEG, MSM25350
4      ETABND,BNDLOC,ETA,EFLUX,FLXSUM,EAVG,LL,RPP,FKP, MSM25360
5      IEMAX,THRSH,TSHENG,TSHING,IRDK,INRGDK,ISOLDK,IONDK, MSM25370
6      DKTIME)          MSM25380
C      VERSION 1.00          DATE: 01.07.88          MSM25390
C      1.10          09.12.89          MSM25400
C      1.20          10.10.89          MSM25410
C      1.30          05.09.90          MSM25420
C      PURPOSE: SUBROUTINE TO TRACE PARTICLE TRAJECTORIES FROM EACH GRID MSM25430
C      POINT FOR EACH PARTICLE SPECIES BACK IN TIME FROM MSM25440
C      TIME=TSTART TO TIME=TSTOP. LOSS BY PRECIPITATION INTO MSM25450
C      THE LOSS CONE IS CALCULATED FOR EACH TEST PARTICLE ALONG MSM25460
C      ITS TRAJECTORY. PARTICLE DISTRIBUTIONS AND PRECIPITATING MSM25470
C      MSM25480
C      MSM25490
C      MSM25500
C      MSM25510
C      MSM25520
C      MSM25530
C      MSM25540
C      MSM25550
C      MSM25560
C      MSM25570
C      MSM25580
C      MSM25590
C      MSM25600
```

90/06/30
12:03:44

msml.for

41

```
C      ENERGY FLUXES ARE CALCULATED AT EACH GRID POINT FOR TIME      MSM25610
C      TSTART FROM INITIAL AND BOUNDARY VALUES AT TIME TSTOP.      MSM25620
C      NOTE: TSTART.GT.TSTOP      MSM25630
C      MSM25640
C      INPUT:      MSM25650
C      TSTART      TIME AT WHICH TO BEGIN PARTICLE TRACEBACK      MSM25660
C      TSTOP      TIME AT WHICH TO END PARTICLE TRACEBACK      MSM25670
C      DTNOM      NOMINAL TIME STEP TO BEGIN TRACES.  DTNOM.LT.0      MSM25680
C      TO TRACE BACK IN TIME.  PPTM WILL ADJUST      MSM25690
C      TIME STEP AS NEEDED.      MSM25700
C      SCTFRFC      SCATTERING FRACTION OF STRONG PITCH ANGLE DIFFUSION      MSM25710
C      IFLAV      SPECIES DEFINITION INDEX.  IFLAV=1 FOR ELECTRONS      MSM25720
C      IFLAV=2 FOR H+ IONS; IFLAV=3 FOR O+ IONS      MSM25730
C      BNDMAX      MAX I VALUE OF BNDY IN TIME INTERVAL      MSM25740
C      ALPHA      LATITUDINAL GRID SPACING VECTOR      MSM25750
C      BETA      LOCAL TIME GRID SPACING VECTOR      MSM25760
C      BIR      IONSOPHERIC MAGNETIC FIELD STRENGTH      MSM25770
C      ITMMAX      NUMBER OF TEMPORAL GRID POINTS USED      MSM25780
C      IDIM      NUMBER OF LATITUDINAL GRID POINTS      MSM25790
C      JDIM      NUMBER OF LOCAL TIME GRID POINTS (INCL WRAPAROUND)      MSM25800
C      IEDIM      NUMBER OF ENERGY SPECIES      MSM25810
C      ITMDIM      MAX NUMBER OF TEMPORAL GRID POINTS      MSM25820
C      IRDIM      R DEPTH IN FLXMAT ARRAY      MSM25830
C      NRGDIM      ENERGY DEPTH IN FLXMAT ARRAY      MSM25840
C      KPDIM      KP DEPTH IN FLXMAT ARRAY      MSM25850
C      KPPLUS      AUGMENTED KP DIMENSION OF FLXMAT (KPDIM+2)      MSM25860
C      ISPDIM      NUMBER OF MASS SPECIES IN FLXMAT ARRAY      MSM25870
C      FLXR      ALOG10(R) VALUES FOR WHICH FLXMAT ARRAY IS      MSM25880
C      CALCULATED      MSM25890
C      FLXNRG      ALOG10(ENERGY) VALUES FOR WHICH FLXMAT ARRAY IS      MSM25900
C      CALCULATED      MSM25910
C      FLXKP      KP VALUES FOR WHICH FLXMAT ARRAY IS CALCULATED      MSM25920
C      FLXMAT      ALOG10(FLUX) FOR A GIVEN R,ENERGY, AND KP      MSM25930
C      THRMAT      ARRAY GIVING LOG OF THRESHOLD FLUXES      MSM25940
C      TIMTAG      VECTOR GIVING UNIVERSAL TIMES FOR WHICH DATA ARRAYS      MSM25950
C      ARE GIVEN. (SECONDS FROM BEGINNING OF START DAY)      MSM25960
C      ALAM      ENERGY INVARIANT OF PARTICLE      MSM25970
C      ALMDEL      ENERGY INVARIANT SPACING      MSM25980
C      VM      (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3      MSM25990
C      R      DISTANCE TO BMIN PT (RE)      MSM26000
C      P      HOUR ANGLE (EASTWARD FROM NOON) (RADIAN)      MSM26010
C      VMLOSS      AUX. LOSS ARRAY= (SINI/BIR)*VM**2      MSM26020
C      V      ELECTRIC POTENTIAL ARRAY (VOLTS)      MSM26030
C      ETABEG      INITIAL ETA DISTRIBUTION (IE., AT T=TSTOP)      MSM26040
C      ETABND      ETA DISTRIBUTION ON BOUNDARY (DEPENDS ON TIME)      MSM26050
C      BNDLOC      OUTER BNDY OF CALCULATION (DEPENDS ON TIME)      MSM26060
C      LL      CURRENT TEMPORAL GRID INDEX      MSM26070
C      RPP      NOMINAL RADIUS OF PLASMAPAUSE      MSM26080
C      FKP      CURRENT KP VALUE      MSM26090
C      IEMAX      NUMBER OF ENERGY CHANNELS BEING USED      MSM26100
C      THRSH      VECTOR GIVING FULL TRACE/DEFAULT FLUX INFORMATION      MSM26110
C      TSHENG      THRESHOLD ENERGY FOR SWITCHING FROM FULL TRACE      MSM26120
C      TO EMPIRICAL MODEL FOR ELECTRONS      MSM26130
C      TSHING      THRESHOLD ENERGY FOR SWITCHING FROM FULL TRACE TO      MSM26140
C      EMPIRICAL MODEL FOR IONS      MSM26150
C      IRDK      R DIMENSION FOR DKTIME ARRAY      MSM26160
C      INRGDK      ENERGY DIMENSION FOR DKTIME ARRAY      MSM26170
C      ISOLDK      SUN SPOT NUMBER DIMENSION FOR DKTIME ARRAY      MSM26180
C      IONDK      NUMBER OF ION SPECIES IN DKTIME ARRAY      MSM26190
C      DKTIME      TABLE OF ION CHARGE EXCHANGE DECAY TIMES      MSM26200
C      MSM26210
C      OUTPUT:      MSM26220
C      ETA      COMPUTED ETA DISTRIBUTION CORRESPONDING TO T=TSTART      MSM26230
C      EFLUX      PRECIPITATING ENERGY FLUX FOR EACH SPECIES      MSM26240
```

42

C	FLXSUM	TOTAL PRECIPITATING ENERGY FLUX FOR: (1) ELECTRONS; (2) H+ IONS; AND (3) O+ IONS	MSM26250
C			MSM26260
C	EAVG	AVERAGE ENERGY OF PRECIPITATING PARTICLES FOR: (1) ELECTRONS; (2) H+ IONS; AND (3) O+ IONS	MSM26270
C			MSM26280
C			MSM26290
C	PROGRAMMER: R. W. SPIRO		MSM26300
C			MSM26310
C			MSM26320
C			MSM26330
C			MSM26340
	PARAMETER (IKDIM=3, IIDIM=62, JJDIM=51, KKDIM=30, MDIM=1000)		MSM26350
	COMMON /GRID/ DLAM, DPSI, RI, JWRAP		MSM26360
	COMMON /LUNIT/LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,		MSM26370
	1 LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,		MSM26380
	2 LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,		MSM26390
	3 LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH		MSM26400
	COMMON /MOVE/ IBEG, ISTOP, JBEG, JSTOP, IEBEG, IEEND		MSM26410
	COMMON /SUN/ SSN		MSM26420
	COMMON/TMP6/III, JJJ		MSM26430
C			MSM26440
	DIMENSION ALPHA (IDIM), BETA (IDIM), BIR (IDIM, JDIM)		MSM26450
	DIMENSION V (IDIM, JDIM, ITMDIM), VM (IDIM, JDIM, ITMDIM), TIMTAG (ITMDIM)		MSM26460
	DIMENSION R (IDIM, JDIM, ITMDIM), P (IDIM, JDIM, ITMDIM)		MSM26470
	DIMENSION IFLAV (IEDIM), BNDLOC (JDIM, ITMDIM), THRSH (IEDIM)		MSM26480
	DIMENSION SCTFRC (IEDIM, JDIM, ITMDIM), RPP (ITMDIM)		MSM26490
	DIMENSION BNDMAX (JDIM)		MSM26500
	DIMENSION VMLOSS (IDIM, JDIM, ITMDIM), ETABEG (IDIM, JDIM, IEDIM)		MSM26510
	DIMENSION ETABND (JDIM, IEDIM, ITMDIM), ETA (IDIM, JDIM, IEDIM)		MSM26520
	DIMENSION EFLUX (IDIM, JDIM, IEDIM)		MSM26530
	DIMENSION FLXSUM (IDIM, JDIM, IKDIM), EAVG (IDIM, JDIM, IKDIM)		MSM26540
	DIMENSION SUM1 (IKDIM), SUM2 (IKDIM)		MSM26550
	DIMENSION XMASS (IKDIM)		MSM26560
	DIMENSION ALAM (IEDIM), ALMDEL (IEDIM)		MSM26570
C			MSM26580
	DIMENSION FLXRFF (IIDIM, JJDIM, KKDIM)		MSM26590
	DIMENSION DENSUM (IIDIM, JJDIM, IKDIM)		MSM26600
C			MSM26610
	DIMENSION FLXMAT (IRDIM, NRGDIM, KPPLUS, ISPDIM)		MSM26620
	DIMENSION THRMAT (IRDIM, NRGDIM, KPDIM)		MSM26630
	DIMENSION FLXR (IRDIM), FLXNRG (NRGDIM), FLXKP (KPDIM)		MSM26640
C			MSM26650
	DIMENSION TM (MDIM), BIM (MDIM), BJM (MDIM), WEAK (MDIM), STRONG (MDIM)		MSM26660
	DIMENSION RR (MDIM), PP (MDIM), VVM (MDIM)		MSM26670
C			MSM26680
	DATA XMASS /9.11E-31, 1.67E-27, 2.67E-26/		MSM26690
	DATA EPS, BBISCL, BBJSCL /.01, 1., 1./		MSM26700
	DATA VMGEO/7.0/		MSM26710
C			MSM26720
C			MSM26730
C			MSM26740
C	CHECK DIMENSION COMPATIBILITY		MSM26750
	IF (IIDIM.NE.IDIM.OR.JJDIM.NE.JDIM.OR.KKDIM.LT.IEMAX) THEN		MSM26760
	WRITE (6, *) 'DIMENSION COMPATIBILITY PROBLEMS IN PPTM'		MSM26770
	WRITE (6, *) 'IIDIM= ', IIDIM, ' SHOULD BE EQUAL TO IDIM= ', IDIM		MSM26780
	WRITE (6, *) 'JJDIM= ', JJDIM, ' SHOULD BE EQUAL TO JDIM= ', JDIM		MSM26790
	WRITE (6, *) 'IEMAX= ', IEMAX, ' SHOULD BE LESS THAN KKDIM= ', KKDIM		MSM26800
	WRITE (6, *) 'STOPPING PGM IN PPTM'		MSM26810
	STOP		MSM26820
	END IF		MSM26830
C			MSM26840
C			MSM26850
	PI=ATAN2(0., -1.)		MSM26860
C			MSM26870
C			MSM26880

C	CALCULATE REFERENCE FLUX ARRAY FOR TIME=TSTART	MSM26890
	CALL SETREF (IDIM,JDIM,IEDIM,ITMDIM,ITMMAX,IRDIM,NRGDIM,KPDIM,	MSM26900
2	KPPLUS,ISPDIM,TIMTAG,TSTART,IFLAV,VM,R,P,BNDLOC,ALAM,	MSM26910
3	ALMDEL,FLXMAT,FLXR,FLXNRG,FLXKP,FLXREF,IEMAX)	MSM26920
C		MSM26930
C	DEBUG PRINTOUT	MSM26940
C	DO 400 IE=1,IEMAX	MSM26950
C	WRITE(6,*) 'IE= ',IE	MSM26960
C	CALL OUTP (FLXREF(1,1,IE),IDIM,JDIM,20,IDIM,1,3,JDIM-1,1,0.,	MSM26970
C	2 'FLXREF',6,132)	MSM26980
C	400 CONTINUE	MSM26990
C		MSM27000
C		MSM27010
C		MSM27020
C		MSM27030
	JMID=(JDIM-3)/2+3	MSM27040
	ETAEQK = 0.	MSM27050
C		MSM27060
C	INITIALIZE FLXSUM AND DENSUM	MSM27070
	DO 100 I=1,IDIM	MSM27080
	DO 100 J=1,JDIM	MSM27090
	DO 110 IK=1,IKDIM	MSM27100
C	IK=1 FOR ELECTRONS; IK=2 FOR H+ IONS; IK=3 FOR O+ IONS	MSM27110
	FLXSUM(I,J,IK)=0.	MSM27120
	DENSUM(I,J,IK)=0.	MSM27130
	110 CONTINUE	MSM27140
C		MSM27150
C	INITIALIZE EFLUX	MSM27160
	DO 120 IE=1,IEMAX	MSM27170
	EFLUX(I,J,IE)=0.	MSM27180
	120 CONTINUE	MSM27190
C		MSM27200
	100 CONTINUE	MSM27210
C		MSM27220
C		MSM27230
C	DO 10 LOOP GOES OVER PARTICLE SPECIES TO BE TRACED	MSM27240
C		MSM27250
	DO 10 IE=IEBEG,IEEND	MSM27260
C		MSM27270
C	FOR ELECTRONS WITH ENERGY GREATER THAN TSHENG KEV AT	MSM27280
C	GEOSYNCHRONOUS ORBIT, USE EMPIRICAL FLUXES AND DISREGARD FULL	MSM27290
C	TRACEBACKS. SIMILARLY FOR IONS GREATER THAN TSHING KEV	MSM27300
C		MSM27310
	HIENG=(ABS (ALAM (IE)) *VMGEO)/1000.0	MSM27320
	IF ((HIENG.GE.TSHENG).AND.(IFLAV(IE).EQ.1)) THEN	MSM27330
	CALL FLXDFL (LL,FKP,R,P,ALAM,VM,IRDIM,NRGDIM,KPDIM,KPPLUS,	MSM27340
1	ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP,EFLUX,ETA,ALMDEL,IDIM,JDIM,	MSM27350
2	IEDIM,ITMDIM,BNDLOC,IEMAX,IFLAV,IE,IE)	MSM27360
	THRSH(IE)=TSHENG	MSM27370
	IEUSE=IE	MSM27380
	GO TO 10	MSM27390
	ENDIF	MSM27400
C		MSM27410
	IF ((HIENG.GE.TSHING).AND.(IFLAV(IE).NE.1)) THEN	MSM27420
	CALL FLXDFL (LL,FKP,R,P,ALAM,VM,IRDIM,NRGDIM,KPDIM,KPPLUS,	MSM27430
1	ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP,EFLUX,ETA,ALMDEL,IDIM,JDIM,	MSM27440
2	IEDIM,ITMDIM,BNDLOC,IEMAX,IFLAV,IE,IE)	MSM27450
	THRSH(IE)=TSHING	MSM27460
	IEUSE=IE	MSM27470
	GO TO 10	MSM27480
	ENDIF	MSM27490
C		MSM27500
	IK=IFLAV(IE)	MSM27510
	ISP=IK	MSM27520

90/06/30
12:03:44

msml.for

44

```
C
C      TIMING INFO FOR TOTAL K VALUE COMP. TIME
C
C      CALL DTIME(ITKST)
C
C      IEUSE=IE
C      AALAM=AALAM(IE)
C
C DO 20 LOOP GOES OVER GRID POINTS
C
C      DO 20 I=IBEG,ISTOP
C          II=1
C          CALL DTIME(ITRCST)
C      DO 24 J=JBEG,JSTOP
C          III=I
C          JJJ=J
C          JJ=J
C          BBT=TNORML(TSTART,TIMTAG,ITMMAX)
C          IBT=BBT
C          BNDJT=(1.-BBT+REAL(IBT))*BNDLOC(JJ,IBT)
1      + (BBT-REAL(IBT))*BNDLOC(JJ,IBT+1)
C
C          IF (REAL(I).LT.BNDJT) THEN
C              ETA(I,J,IE)=(1.-BBT+REAL(IBT))*ETABND(J,IE,IBT)
1          + (BBT-REAL(IBT))*ETABND(J,IE,IBT+1)
C              GO TO 24
C          END IF
C
C      TIME LOOP HAS POSSIBILITY FOR VARYING TIME STEPS. STOP TRACING
C      PARTICLE TRAJECTORY WHEN (1) BOUNDARY IS CROSSED OR (2) TSTOP IS
C      REACHED.
C
C      INITIALIZATION
C
C          T=TSTART
C          IT=NINT(T)
C          DTNEXT=DTNOM
C          FXLOSS=0.
C          FXL2=0.
C          BBI=I
C          BBJ=J
C          BBT=TNORML(T,TIMTAG,ITMMAX)
C          M=1
C          TM(M)=T
C          BIM(M)=BBI
C          BJM(M)=BBJ
C          RR(M)=G3NTRP(R,IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
C          PP(M)=G3TRPA(P,IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
C          VVM(M)=G3NTRP(VM,IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
C          RPPM=G3NTRP(RPP,ITMDIM,1,1,BBT,1.,1.)
C          ENRGM=ABS(AALAM)*VVM(M)
C
C      IF (ISP.EQ.1) THEN
C          FZERO=G3NTRP(SCTFRC,IEDIM,JDIM,ITMDIM,FLOAT(IE),BBJ,BBT)
C          RATEF=(4.45E-17)*SQRT(ABS(AALAM)/XMASS(1K))*G3NTRP(VMLOSS,
2          IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
C          STRONG(M)=FZERO*RATEF
C          WEAK(M)=WKRATE(RR(M),RPPM,PP(M),ENRGM)
C      ELSE
C          STRONG(M)=0.
C          WEAK(M)=CEXRAT(ISP,ENRGM,RR(M),SSN,DKTIME,IRDK,INRGDK,
2          ISOLDK,IONDK)
C      END IF
```

MSM27530
MSM27540
MSM27550
MSM27560
MSM27570
MSM27580
MSM27590
MSM27600
MSM27610
MSM27620
MSM27630
MSM27640
MSM27650
MSM27660
MSM27670
MSM27680
MSM27690
MSM27700
MSM27710
MSM27720
MSM27730
MSM27740
MSM27750
MSM27760
MSM27770
MSM27780
MSM27790
MSM27800
MSM27810
MSM27820
MSM27830
MSM27840
MSM27850
MSM27860
MSM27870
MSM27880
MSM27890
MSM27900
MSM27910
MSM27920
MSM27930
MSM27940
MSM27950
MSM27960
MSM27970
MSM27980
MSM27990
MSM28000
MSM28010
MSM28020
MSM28030
MSM28040
MSM28050
MSM28060
MSM28070
MSM28080
MSM28090
MSM28100
MSM28110
MSM28120
MSM28130
MSM28140
MSM28150
MSM28160

```
C
C
C  DEBUG 1.5.90
C  WRITE (6,789)
C 789 FORMAT(1X,'*****')
C  WRITE (6,790) M,TM(M),BIM(M),BJM(M),RR(M),PP(M),VVM(M),RPPM,ENRGM
C 790 FORMAT(1X,'M=',I3,2X,'T=',F8.0,2X,'BI=',F6.3,2X,'BJ=',F6.3,2X,
C 2  'R=',F6.3,2X,'P=',F8.3,2X,'VM=',F8.3,2X,'RPP=',F6.3,2X,
C 3  'ENRG=',1PE12.3)
C  WRITE (6,791) M,T,RATEF,STRONG(M),WEAK(M)
C 791 FORMAT(1X,'DEBUG 1.5.90  M=',I5,2X,'T=',F10.0,2X,
C 2  'RATEF=',1PE12.3,2X,'STRONG LOSS RATE=',E12.3,2X,
C 3  'WEAK LOSS RATE=',E12.3)
C
C  BEGINNING OF TIME LOOP
C
C 30 CONTINUE
C
C  MOVE PARTICLE A TIME STEP
C
C      TOLD=T
C      BIOLD=BBI
C      BJOLD=BBJ
C      DTTRY=DTNEXT
C
C      IF (DTNOM.GT.0..AND.DTTRY.GT.(TSTOP-TOLD)) DTTRY=TSTOP-TOLD
C      IF (DTNOM.LT.0..AND.DTTRY.LT.(TSTOP-TOLD)) DTTRY=TSTOP-TOLD
C
C      IF (BIOLD.GT.REAL(IDIM)-1.) THEN
C          ETA(I,J,IE)=0.
C          GO TO 24
C      END IF
C      CALL MOVER(T,DTTRY,BIOLD,BJOLD,EPS,BBISCL,BBJSCL,DTUSED,DTNEXT,
C 1      AALAM,V,VM,TIMTAG,ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM,BBI,BBJ)
C
C  DEBUG PRINTOUT 12.29.89
C  WRITE (6,888) IE,T,DTTRY,BIOLD,BJOLD,DTUSED,BBI,BBJ
C 888 FORMAT(1X,'IE=',I2,2X,'T=',F10.2,2X,'DTTRY=',F5.0,2X,'BIOLD=',
C 2  F6.2,2X,'BJOLD=',F6.2,2X,'DTUSED=',F5.0,2X,'NEW BI=',F6.2,
C 3  2X,'NEW BJ=',F6.2)
C
C
C
C
C      IT=IT+NINT(DTUSED)
C      T=REAL(IT)
C      BBT=TNORML(T,TIMTAG,ITMMAX)
C
C
C      M=M+1
C      BIM(M)=BBI
C      BJM(M)=BBJ
C      CALL MODFIX(BJM(M),JDIM,JWRAP)
C      TM(M)=T
C
C
C      RR(M)=G3NTRP(R,IDIM,JDIM,ITMDIM,BIM(M),BJM(M),BBT)
C      PP(M)=G3TRPA(P,IDIM,JDIM,ITMDIM,BIM(M),BJM(M),BBT)
C      VVM(M)=G3NTRP(VM,IDIM,JDIM,ITMDIM,BIM(M),BJM(M),BBT)
C      RPPM=G3NTRP(RPP,ITMDIM,1,1,BBT,1.,1.)
C
C
C
C
C      ENRG=ABS(ALAM(IE))*VVM(M)
C      FZERO=G3NTRP(SCTFRC,IEDIM,JDIM,ITMDIM,FLOAT(IE),BBJ,BBT)
```

MSM28170
MSM28180
MSM28190
MSM28200
MSM28210
MSM28220
MSM28230
MSM28240
MSM28250
MSM28260
MSM28270
MSM28280
MSM28290
MSM28300
MSM28310
MSM28320
MSM28330
MSM28340
MSM28350
MSM28360
MSM28370
MSM28380
MSM28390
MSM28400
MSM28410
MSM28420
MSM28430
MSM28440
MSM28450
MSM28460
MSM28470
MSM28480
MSM28490
MSM28500
MSM28510
MSM28520
MSM28530
MSM28540
MSM28550
MSM28560
MSM28570
MSM28580
MSM28590
MSM28600
MSM28610
MSM28620
MSM28630
MSM28640
MSM28650
MSM28660
MSM28670
MSM28680
MSM28690
MSM28700
MSM28710
MSM28720
MSM28730
MSM28740
MSM28750
MSM28760
MSM28770
MSM28780
MSM28790
MSM28800

```
C
C  CALCULATE LOSS RATE THIS TIME STEP
C
C      IF (ISP.EQ.1) THEN
C
C          CALCULATE LOSS RATE FOR ELECTRONS
C
C      FIRST, CALCULATE STRONG PITCH ANGLE SCATTERING RATE
C          RATEF=(4.45E-17)*SQRT (ABS (AALAM) /XMASS (IK) ) *G3NTRP (VMLOSS,
C      2          IDIM,JDIM,ITMDIM,BBI,BBJ,BBT)
C          STRONG (M) =FZERO*RATEF
C
C      NEXT, CALCULATE WEAK LOSS RATE
C          WEAK (M) =WKRATE (RR (M) ,RPPM,PP (M) ,ENRG)
C
C      ELSE
C
C          CALCULATE LOSS RATE FOR IONS
C          STRONG IS SET TO ZERO FOR IONS
C          WEAK IS USED AS AN ALIAS FOR CHARGE EXCHANGE LOSS RATE
C          SSN IS SUNSPOT NUMBER
C          ISP IS SPECIES NUMBER  (2 FOR H+, 3 FOR O+)
C
C          WEAK (M) =CEXRAT (ISP,ENRG,RR (M) ,SSN,DKTIME,IRDK,INRGDK,
C      2          ISOLDK,IONDK)
C
C          STRONG (M) =0.
C
C      END IF
C
C      CORRECT J MODULUS OF BBJ
C
C          CALL MODFIX (BBJ,JDIM,JWRAP)
C
C      CHECK IF PARTICLE HAS PASSED EITHER TIME OR SPATIAL BOUNDARY
C      IF SO, EXIT TIME LOOP
C
C          IEXIT=0
C
C      IF TIME BOUNDARY IS CROSSED, SET IEXIT=1 AND INTERPOLATE
C      TO FIND ETABEG AT LOCATION (BBI,BBJ)
C
C      25 CONTINUE
C          CALL TCHK (T,TSTOP,DTNOM,BBI,BBJ,IEUSE,ETABEG,IDIM,JDIM,
C      1          IEDIM,ETANIL,IEXIT)
C
C          IF (IEXIT.EQ.0) THEN
C
C              CHECK TO SEE IF SPATIAL BNDY IS CROSSED
C
C              CALL BNDCHK (T,IEUSE,BBI,BBJ,BIOLD,BJOLD,TIMTAG,ETABND,
C      1          BNDMAX,BNDLOC,JDIM,IEDIM,ITMDIM,ITMMAX,JWRAP,ETANIL,IEXIT)
C
C          END IF
C
C          IF (IEXIT.EQ.0) THEN
```

```
MSM28810
MSM28820
MSM28830
MSM28840
MSM28850
MSM28860
MSM28870
MSM28880
MSM28890
MSM28900
MSM28910
MSM28920
MSM28930
MSM28940
MSM28950
MSM28960
MSM28970
MSM28980
MSM28990
MSM29000
MSM29010
MSM29020
MSM29030
MSM29040
MSM29050
MSM29060
MSM29070
MSM29080
MSM29090
MSM29100
MSM29110
MSM29120
MSM29130
MSM29140
MSM29150
MSM29160
MSM29170
MSM29180
MSM29190
MSM29200
MSM29210
MSM29220
MSM29230
MSM29240
MSM29250
MSM29260
MSM29270
MSM29280
MSM29290
MSM29300
MSM29310
MSM29320
MSM29330
MSM29340
MSM29350
MSM29360
MSM29370
MSM29380
MSM29390
MSM29400
MSM29410
MSM29420
MSM29430
MSM29440
```

90/06/30
12:03:44

msml.for

47

```

      GO TO 30
C      FOR NEXT TIME STEP
      ELSE
C      CALCULATE ETA(I,J,IE)
C
C      DO 300 N=M,2,-1
C
C      CALCULATE EQUILIBRIUM ETA AT PT (N-1)
      BT=TNORML(TM(N-1),TIMTAG,ITMMAX)
      BIT=BIM(N-1)
      BJT=BJM(N-1)
      BIE=FLOAT(IE)
      ENRG=ABS(ALAM(IE))*VVM(N-1)
      FKPEQ=-(KPDIM+1)
      FLXEQK=FLXVAL(ISP,FKPEQ,RR(N-1),PP(N-1),ENRG,IRDIM,
2          NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,
3          FLXNRG,FLXKP)
C
C      IF(ISP.EQ.1) THEN
C
C          ETAEQK=FLXEQK*ALMDEL(IE)/7.392E-16/
2          Sqrt(ABS(ALAM(IE)))/VVM(N-1)
C
C      ELSE
      IF(ISP.EQ.2) AWT=1.
      IF(ISP.EQ.3) AWT=16.
      ETAEQK=(FLXEQK*ALMDEL(IE)*Sqrt(AWT))/(1.731E-17*
2          Sqrt(ABS(ALAM(IE))))/VVM(N-1)
C
C      END IF
C
C
C
C      IF(ISP.EQ.1) THEN
C          CALCULATE THRESHOLD ETA
C          SET UP INTERPOLATION PARAMETERS FOR THRMAT ARRAY
      BKP=FKP+1.
      IF(BKP.GT.7.) BKP=7.
      BNRG=4.*ALOG10(ENRG)-3.
C
      RLOG=ALOG10(RR(N-1))
      IF(RLOG.GE.FLXR(IRDIM)) THEN
      BR=IRDIM
      THRLOG=G3NTRP(THRMAT,IRDIM,NRGDIM,KPDIM,BR,BNRG,
2          BKP)+ALOG10(13./RR(N-1))
C
      ELSE
C
      DO 18 IR=1,IRDIM-1
      IF(RLOG.LE.FLXR(IR+1)) THEN
      BR=FLOAT(IR)+(RLOG-FLXR(IR))/(FLXR(IR+1)-
2          FLXR(IR))
      GO TO 19
      END IF
18      CONTINUE
C
19      CONTINUE
C
      THRLOG=G3NTRP(THRMAT,IRDIM,NRGDIM,KPDIM,BR,BNRG,
2          BKP)
C
C      END IF
C
C      THRSHJ=10.**THRLOG
C
```

MSM29450
MSM29460
MSM29470
MSM29480
MSM29490
MSM29500
MSM29510
MSM29520
MSM29530
MSM29540
MSM29550
MSM29560
MSM29570
MSM29580
MSM29590
MSM29600
MSM29610
MSM29620
MSM29630
MSM29640
MSM29650
MSM29660
MSM29670
MSM29680
MSM29690
MSM29700
MSM29710
MSM29720
MSM29730
MSM29740
MSM29750
MSM29760
MSM29770
MSM29780
MSM29790
MSM29800
MSM29810
MSM29820
MSM29830
MSM29840
MSM29850
MSM29860
MSM29870
MSM29880
MSM29890
MSM29900
MSM29910
MSM29920
MSM29930
MSM29940
MSM29950
MSM29960
MSM29970
MSM29980
MSM29990
MSM30000
MSM30010
MSM30020
MSM30030
MSM30040
MSM30050
MSM30060
MSM30070
MSM30080

90/06/30
12:03:44

msm1.for

48

```
C
      2      ETATHR=THRSJ*ALMDEL(IE)/7.392E-16/
              Sqrt (ABS (ALAM(IE)))/VVM(N-1)
C
C
C
      END IF
C
C
C  CALCULATE RHS OF D(ETA)/DT EQN
      WKMID=.5*(WEAK(N)+WEAK(N-1))
      STRMID=.5*(STRONG(N)+STRONG(N-1))
      DETDTW=AMAX1(0.,(ETANIL-ETAEQK))*WKMID
C
      DETDTS=AMAX1(0.,(ETANIL-ETATHR))*STPMID
C
C
C
      IF (DETDTW.EQ.0..AND.DETDTS.EQ.0.) THEN
        ETANIL=ETANIL
        RATE=0.
        GO TO 300
      END IF
C
      IF (DETDTW.GT.DETDTS) THEN
        RATE=WKMID
        ETAM=ETAEQK
        RATE=DETDTW
      ELSE
        RATE=STRMID
        ETAM=ETATHR
        RATE=DETDTS
      END IF
C
      IF (RATE.GT.10.) THEN
        WRITE(6,*) 'I,J,IE,WKMID,STRMID',I,J,IE,WKMID,STRMID
        WRITE(6,795) DETDTW,DETDTS,ETANIL
795  FORMAT(1X,'DETDW=',1PE12.3,2X,'DETDTS=',E12.3,2X,'ETA(N)=' ,E12.3)
        WRITE(6,799) N,ETAM,ETANIL,TM(N-1),TM(N),RATE
799  FORMAT(1X,'N=',I6,2X,'ETAM=',1PE12.3,2X,'ETANIL=' ,E12.3,2X,
2      'TM(N-1)=' ,F10.2,2X,'TM(N)=' ,F10.2,2X,'RATE=' ,E12.3)
        WRITE(6,*) 'STOPPING IN PPTM'
        STOP
      END IF
C
      ETAOLD=ETANIL
      ETANIL=ETAM+(ETAOLD-ETAM)*EXP(-(TM(N-1)-TM(N))*RATE)
C
C
C
C
      300 CONTINUE
C
C
      ETA(I,J,IE)=ETANIL
C
C
      BI=I
      BJ=J
      BT=TNORML(TSTART,TIMTAG,ITMMAX)
      VMVAL=G3NTRP(VM,IDIM,JDIM,ITMDIM,BI,BJ,BT)
      ENRG=ABS(ALAM(IE))*VMVAL
```

MSM30090
MSM30100
MSM30110
MSM30120
MSM30130
MSM30140
MSM30150
MSM30160
MSM30170
MSM30180
MSM30190
MSM30200
MSM30210
MSM30220
MSM30230
MSM30240
MSM30250
MSM30260
MSM30270
MSM30280
MSM30290
MSM30300
MSM30310
MSM30320
MSM30330
MSM30340
MSM30350
MSM30360
MSM30370
MSM30380
MSM30390
MSM30400
MSM30410
MSM30420
MSM30430
MSM30440
MSM30450
MSM30460
MSM30470
MSM30480
MSM30490
MSM30500
MSM30510
MSM30520
MSM30530
MSM30540
MSM30550
MSM30560
MSM30570
MSM30580
MSM30590
MSM30600
MSM30610
MSM30620
MSM30630
MSM30640
MSM30650
MSM30660
MSM30670
MSM30680
MSM30690
MSM30700
MSM30710
MSM30720

```
C C C C COMPUTE PARTICLE FLUX ASSOCIATED WITH ETA(I,J,IE). IF COMPUTED  
C FLUX IS GREATER THAN UPPER LIMIT REFERENCE FLUX, THEN REDUCE ETA  
C TO BE CONSISTENT WITH REFERENCE FLUX. REF: SECTION 2.5.6 OF THE  
C FINAL REPORT FOR CONTRACT #F19628-87-K-0001.  
C  
      IF(ISP.EQ.1) THEN  
        FLUX=7.392E-16*SQR(ABS(ALAM(IE)))*VMVAL*  
    2     ETA(I,J,IE)/ALMDEL(IE)  
      ELSE  
        IF(ISP.EQ.2) THEN  
          AWT=1.  
        ELSE  
          AWT=16.  
        END IF  
  
        FLUX=1.731E-17*SQR(ABS(ALAM(IE)))*VMVAL*  
    2     ETA(I,J,IE)/(ALMDEL(IE)*SQR(AWT))  
  
      END IF  
  
892 FORMAT(1X,'VMVAL=',F8.3,2X,'ALMDEL=',1PE12.3,2X,'FLUX=',E12.3,2X,  
    2   'FLXREF=',E12.3)  
      IF(FLUX.GT.FLXREF(I,J,IE)) THEN  
C       WRITE(6,*) 'COMPUTED FLUX EXCEEDS REFERENCE FLUX',  
C         2   ' I=',I,' J=',J,' IE=',IE,  
C         3   ' FLUX=',FLUX,' REF FLUX=' ,FLXREF(I,J,IE)  
C CORRECT ETA TO REFERENCE FLUX VALUE  
      ETAS=ETA(I,J,IE)  
      ETA(I,J,IE)=ETA(I,J,IE)*FLXREF(I,J,IE)/FLUX  
  
C DEBUG PRINTOUT  
C       WRITE(6,*) 'COMPUTED ETA= ',ETAS,  
C         2   ' CORRECTED ETA=',ETA(I,J,IE)  
  
      END IF  
  
    END IF  
  
END IF  
  
23 CONTINUE  
  
IF(ISP.EQ.1) THEN  
  
CALCULATE PRECIPITATING ENERGY FLUX AND FLUX SUMS  
  BBI=I  
  BBJ=J  
  BIRM=G3NTRP(BIR,IDIM,JDIM,1,BBI,BBJ,1.)  
  DETDTW=AMAX1(0.,(ETA(I,J,IE)-ETAEQK))*WEAK(1)  
  DETDTS=AMAX1(0.,(ETA(I,J,IE)-ETATHR))*STRONG(1)  
  RATETA=AMAX1(DETDTW,DETDTS)  
  FLUXN=RATETA*BIRM/2.E13  
  EFLUX(I,J,IE)=.8E-25*RATETA*BIRM*ABS(AALAM)*VVM(1)  
  FLXSUM(I,J,IK)=FLXSUM(I,J,IK)+EFLUX(I,J,IE)  
  DENSUM(I,J,IK)=DENSUM(I,J,IK)+FLUXN  
  
END IF  
24 CONTINUE  
  ETA(I,1,IE)=ETA(I,JDIM-JWRAP+1,IE)
```

90/06/30
12:03:44

msm1.for

50

```
ETA(I,2,IE)=ETA(I,JDIM-JWRAP+2,IE)
ETA(I,JDIM,IE)=ETA(I,JWRAP,IE)
IF(ISP.EQ.1) THEN
  EFLUX(I,1,IE)=EFLUX(I,JDIM-JWRAP+1,IE)
  EFLUX(I,2,IE)=EFLUX(I,JDIM-JWRAP+2,IE)
  EFLUX(I,JDIM,IE)=EFLUX(I,JWRAP,IE)
  FLXSUM(I,1,IK)=FLXSUM(I,JDIM-JWRAP+1,IK)
  FLXSUM(I,2,IK)=FLXSUM(I,JDIM-JWRAP+2,IK)
  FLXSUM(I,JDIM,IK)=FLXSUM(I,JWRAP,IK)
  DENSUM(I,1,IK)=DENSUM(I,JDIM-JWRAP+1,IK)
  DENSUM(I,2,IK)=DENSUM(I,JDIM-JWRAP+2,IK)
  DENSUM(I,JDIM,IK)=DENSUM(I,JWRAP,IK)
END IF

C
C
C PRINT THE TIME SPENT PROCESSING LATITUDE I
C
C CALL DTIME(ITRCEN)
C WRITE (LUERR,*) 'THE TIME SPENT TRACING PARTICLE ',IE,
C $ 'AT LATITUDE INDEX ',I,' WAS ',
C $ ABS(ITRCEN-ITRCST) ', SECONDS.'
C
C
C
C 20 CONTINUE
C
C
C WRITE(18,*) 'IE=',IE,'T=',TSTART
C WRITE(6,*) 'IE=',IE,'T=',TSTART
C
C
C IBM=IDIM
C DO 6 JJJ=1,JDIM
C   IF(INT(BNDLOC(JJJ,LL)).LT.IBM) IBM=BNDLOC(JJJ,LL)
C 6 CONTINUE
C
C
C
C PRINT TOTAL COMP. TIME FOR THIS K VALUE
C
C CALL DTIME(ITKEND)
C WRITE (LUERR,*) 'THE TOTAL TIME SPENT TRACING K = ', IE,
C $ ' WAS ', ABS(ITKEND-ITKST) ', SECONDS.'
C
C 10 CONTINUE
C
C
C
C CALCULATE PRECIPITATING FLUXES AND ENERGIES
C
C   INITIALIZATION OF PRECIPITATION ARRAYS
C
C
C DO 200 I=IBEG,ISTOP
C DO 200 J=JBEG,JSTOP
C   IF(DENSUM(I,J,1).GT.0.) THEN
C     EAVG(I,J,1)=FLXSUM(I,J,1)/DENSUM(I,J,1)
C 200 CONTINUE
C   ELSE
C     EAVG(I,J,1)=0.
C   END IF
C
C 1 EAVG(I,J,1)
C
C 200 CONTINUE
```

MSM31370
MSM31380
MSM31390
MSM31400
MSM31410
MSM31420
MSM31430
MSM31440
MSM31450
MSM31460
MSM31470
MSM31480
MSM31490
MSM31500
MSM31510
MSM31520
MSM31530
MSM31540
MSM31550
MSM31560
MSM31570
MSM31580
MSM31590
MSM31600
MSM31610
MSM31620
MSM31630
MSM31640
MSM31650
MSM31660
MSM31670
MSM31680
MSM31690
MSM31700
MSM31710
MSM31720
MSM31730
MSM31740
MSM31750
MSM31760
MSM31770
MSM31780
MSM31790
MSM31800
MSM31810
MSM31820
MSM31830
MSM31840
MSM31850
MSM31860
MSM31870
MSM31880
MSM31890
MSM31900
MSM31910
MSM31920
MSM31930
MSM31940
MSM31950
MSM31960
MSM31970
MSM31980
MSM31990
MSM32000

90/06/30
12:03:44

msml.for

51

```
C
C
C
C
C      RETURN
C      END
C      SUBROUTINE BNDCHK(T,IE,BBI,BBJ,BIOLD,BBJOLD,TIMTAG,ETABND,BNDMAX,
1      BNDLOC,JDIM,IEDIM,ITMDIM,ITMMAX,JWRAP,ETANIL,IEXIT)
C
C      VERSION 1.00                DATE: 01.11.88
C
C      PURPOSE:  SUBROUTINE TO DETERMINE WHETHER PARTICLE HAS CROSSED
C                 BNDY OF CALCULATION.  IF SO, IEXIT IS SET EQUAL TO 2 AND
C                 ETANIL=ETA VALUE AT BOUNDARY CROSSING PT IS CALCULATED.
C
C      INPUT:
C          T          TIME
C          IE         PARTICLE SPECIES (IE., ENERGY CHANNEL)
C          BBI        I LOCATION AT END OF TIME STEP
C          BBJ        J LOCATION AT END OF TIME STEP
C          BIOLD       I LOCATION AT BEGINNING OF TIME STEP
C          BBJOLD      J LOCATION AT BEGINNING OF TIME STEP
C          TIMTAG      VECTOR GIVING EVENT TIMES FOR WHICH DATA ARRAYS ARE
C                     GIVEN
C          ETABND      ETA DISTRIBUTION ON BOUNDARY (DEPENDS ON TIME)
C          BNDMAX      VECTOR GIVING MAX I VALUE OF BNDY LOCATION AT EACH
C                     J VALUE
C          BNDLOC      ARRAY GIVING BNDY LOCATION AS FUNCTION OF TIME
C          JDIM        NUMBER OF LONGITUDE GRID POINTS (INCL WRAPAROUND)
C          IEDIM       NUMBER OF ENERGY SPECIES
C          ITMDIM      MAX NUMBER OF TEMPORAL GRID POINTS
C          ITMMAX      NUMBER OF TEMPORAL PTS FILLED
C          JWRAP       J WRAP-AROUND
C
C      OUTPUT:
C          ETANIL      IF BNDY IS CROSSED, ETANIL IS THE VALUE OF ETA
C                     AT THE BNDY CROSSING PT
C          IEXIT       IF BNDY IS CROSSED, IEXIT= 2
C
C      PROGRAMMER: R. W. SPIRO
C
C
C
C
C      DIMENSION TIMTAG(ITMDIM),BNDMAX(JDIM),BNDLOC(JDIM,ITMDIM)
C      DIMENSION ETABND(JDIM,IEDIM,ITMDIM)
C      COMMON/TMP6/III,JJJ
C          COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
C
C
C      IF SPATIAL BOUNDARY IS CROSSED, SET IEXIT=2, CALCULATE CROSSING
C      POINT, AND VALUE OF ETA AT CROSSING POINT
C
C      INITIAL CHECK - EVALUATE BI AT MAXIMUM BOUNDARY VALUES AT BJ=BBJ
C
C          JJ=BBJ
C          BIVAL=(BBJ-REAL(JJ))*BNDMAX(JJ+1)+(REAL(JJ+1)-BBJ)*BNDMAX(JJ)
C
C          IF (BBI.LE.BIVAL) THEN
```

90/06/30
12:03:44

52

msm1.for

```
C                               WE NEED TO PERFORM TIME INTERPOLATION OF MSM32650
C                               BOUNDARY LOCATION AND CHECK AGAINST    MSM32660
C                               LOCATION (BBI,BBJ)                      MSM32670
C                               MSM32680
C                               BBT=TNORML (T, TIMTAG, ITMMAX)          MSM32690
C                               ITM=BBT                                MSM32700
C                               BNDJ=(1.-BBT+REAL (ITM) ) *BNDLOC (JJ, ITM) MSM32710
C                               1      + (BBT-REAL (ITM) ) *BNDLOC (JJ, ITM+1) MSM32720
C                               BNDJP=(1.-BBT+REAL (ITM) ) *BNDLOC (JJ+1, ITM) MSM32730
C                               1      + (BBT-REAL (ITM) ) *BNDLOC (JJ+1, ITM+1) MSM32740
C                               MSM32750
C                               BIVAL=(BBJ-REAL (JJ) ) *BNDJP+ (REAL (JJ+1) -BBJ) *BNDJ MSM32760
C                               MSM32770
C                               IF (BBI.LE.BIVAL) THEN                  MSM32780
C                               MSM32790
C                               PARTICLE HAS CROSSED SPATIAL BOUNDARY    MSM32800
C                               SET IEXIT=2 AND CALCULATE CROSSING POINT  MSM32810
C                               MSM32820
C                               IEXIT=2                                  MSM32830
C                               MSM32840
C                               BJOLD=BBJOLD                             MSM32850
C                               NEED TO GET BBJ AND BJOLD IN SAME MODULUS MSM32860
C                               IF (BBJOLD-BBJ.GT.REAL (JDIM-JWRAP) /2.) THEN MSM32870
C                               BJOLD=BBJOLD-REAL (JDIM-JWRAP)           MSM32880
C                               END IF                                    MSM32890
C                               MSM32900
C                               IF (BBJ-BBJOLD.GT.REAL (JDIM-JWRAP) /2.) THEN MSM32910
C                               BJOLD=BJOLD+REAL (JDIM-JWRAP)           MSM32920
C                               END IF                                    MSM32930
C                               MSM32940
C                               LINE1 IS LINE FROM OLD LOCATION (BIOLD,BJOLD) TO NEW LOCATION MSM32950
C                               (BBI,BBJ). DESCRIBE LINE AS BI=DIDJ*BJ+BICON MSM32960
C                               MSM32970
C                               IF (ABS (BBJ-BJOLD) .LE.1.E-6) THEN      MSM32980
C                               BJCRSS=BBJ                               MSM32990
C                               ELSE                                       MSM33000
C                               DIDJ1=(BBI-BIOLD) / (BBJ-BJOLD)          MSM33010
C                               BICON1=(BIOLD*BBJ-BBI*BJOLD) / (BBJ-BJOLD) MSM33020
C                               JJBEG=MIN (BJOLD, BBJ)                   MSM33030
C                               JJEND=MAX (BJOLD, BBJ)                   MSM33040
C                               MSM33050
C                               CHECK ALL POSSIBLE CROSSING SEGMENTS     MSM33060
C                               MSM33070
C                               DO 40 JJ=JJBEG, JJEND                     MSM33080
C                               BNDJJ=(1.-BBT+REAL (ITM) ) *BNDLOC (JJ, ITM) MSM33090
C                               1      + (BBT-REAL (ITM) ) *BNDLOC (JJ, ITM+1) MSM33100
C                               BNDJJP=(1.-BBT+REAL (ITM) ) *BNDLOC (JJ+1, ITM) MSM33110
C                               1      + (BBT-REAL (ITM) ) *BNDLOC (JJ+1, ITM+1) MSM33120
C                               MSM33130
C                               DIDJ2=(BNDJJP-BNDJJ)                     MSM33140
C                               BICON2=BNDJJ*REAL (JJ+1) -BNDJJP*REAL (JJ) MSM33150
C                               IF (ABS (DIDJ2-DIDJ1) .LE.1.E-6) GO TO 40 MSM33160
C                               MSM33170
C                               BJCRSS=(BICON1-BICON2) / (DIDJ2-DIDJ1)   MSM33180
C                               MSM33190
C                               CHECK IF CROSSING IS BETWEEN JJ AND JJ+1 MSM33200
C                               MSM33210
C                               IF (BJCRSS.GE.REAL (JJ) .AND. BJCRSS.LE.REAL (JJ+1) ) THEN MSM33220
C                               BOUNCE OUT OF DO 40 LOOP                 MSM33230
C                               GO TO 50                                  MSM33240
C                               END IF                                    MSM33250
C                               IF (JJBEG.EQ.JJEND) GO TO 50             MSM33260
C                               40 CONTINUE                               MSM33270
C                               MSM33280
```

90/06/30
12:03:44

msml.for

53

```
C  DEBUG - FAILED TO FIND BOUNDARY CROSSING                      MSM33290
C                                                                MSM33300
C  WRITE(6,*) 'FAILED TO FIND BNDY CROSSING IN SUBROUTINE PPTM'  MSM33310
C  WRITE(6,*) 'T,IE,BBI,BBJ,BIOLD,BBJOLD',T,IE,BBI,BBJ,BIOLD,BBJOLD MSM33320
C  WRITE(6,*) 'BJCRSS',BJCRSS                                    MSM33330
C  BJCRSS=BBJ                                                    MSM33340
C  WRITE(6,*) 'USING BBJ AS CROSSING POINT=',BBJ                MSM33350
C                                                                MSM33360
C                                                                MSM33370
C  50      CONTINUE                                              MSM33380
C                                                                MSM33390
C  CROSSING POINT IS BJCRSS.  COMPUTE ETANIL= ETA AT BNDY CROSSING MSM33400
C                                                                MSM33410
C          JJ=BJCRSS                                             MSM33420
C          ETAJ=(1.-BBT+REAL(ITM))*ETABND(JJ,IE,ITM)             MSM33430
C  1          + (BBT-REAL(ITM))*ETABND(JJ,IE,ITM+1)             MSM33440
C          ETAJP=(1.-BBT+REAL(ITM))*ETABND(JJ+1,IE,ITM)          MSM33450
C  1          + (BBT-REAL(ITM))*ETABND(JJ+1,IE,ITM+1)           MSM33460
C                                                                MSM33470
C          ETANIL=(1.-BJCRSS+REAL(JJ))*ETAJ+(BJCRSS-REAL(JJ))*ETAJP MSM33480
C          IF(IE.EQ.1.OR.IE.EQ.17) WRITE(6,900)III,JJJ,ITM,IE,  MSM33490
C  1          BBT,BBI,BBJ,JJ,ETAJ,ETAJP,ETANIL                  MSM33500
C900      FORMAT(1H,'III,JJJ,ITM,IE,BBT,BBI,BBJ,JJ,ETAJ,ETAJP,  MSM33510
C  1          ETANIL'/4I4,3F15.5,I4,3E15.5)                     MSM33520
C          END IF                                                MSM33530
C                                                                MSM33540
C          END IF                                                MSM33550
C                                                                MSM33560
C          END IF                                                MSM33570
C                                                                MSM33580
C          RETURN                                                MSM33590
C          END                                                    MSM33600
C          FUNCTION DVEFDI(I,J,BT,AALAM,V,VM,IDIM,JDIM,ITMDIM)    MSM33610
C                                                                MSM33620
C  VERSION 1.00                      DATE: 12.16.87              MSM33630
C          1.01A                      02.02.89                  MSM33640
C                                                                MSM33650
C  PURPOSE:  FUNCTION SUBPROGRAM TO COMPUTE THE I DERIVATIVE OF THE MSM33660
C            EFFECTIVE POTENTIAL AT GRID POINT (I,J) AT NORMALIZED MSM33670
C            TIME BT.                                             MSM33680
C                                                                MSM33690
C  INPUT:                                             MSM33700
C  I          I COORDINATE OF THE GRID POINT                  MSM33710
C  J          J COORDINATE OF THE GRID POINT                  MSM33720
C  BT         NORMALIZED TIME                                  MSM33730
C  AALAM      ENERGY INVARIANT (EV (RE/NT)**2/3)             MSM33740
C  V          ELECTRIC POTENTIAL ARRAY (VOLTS)                 MSM33750
C  VM         (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3 MSM33760
C  IDIM       I DIMENSION OF V AND VM                          MSM33770
C  JDIM       J DIMENSION OF V AND VM                          MSM33780
C  ITMDIM     TIME DIMENSION OF V AND VM                       MSM33790
C                                                                MSM33800
C  OUTPUT:                                             MSM33810
C  DVEFDI      D(V+AALAM*VM)/DI AT (I,J,BT)                   MSM33820
C                                                                MSM33830
C  PROGRAMMER:  R. W. SPIRO                                     MSM33840
C                                                                MSM33850
C                                                                MSM33860
C                                                                MSM33870
C  COMMON /GRID/DLAM,DPSI,RI,JWRAP                          MSM33880
C                                                                MSM33890
C  COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,     MSM33900
C  1  LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,    MSM33910
C  2  LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,  MSM33920
```

90/06/30
12:03:44

54

msml.for

```
3      LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH
      DIMENSION V(IDIM,JDIM,ITMDIM),VM(IDIM,JDIM,ITMDIM)
C
      ITT=BT
      FT=REAL(ITT)
      IMIN=1
      IMAX=IDIM
      IF(J.LT.2) THEN
          JX=J+(JDIM-JWRAP)
      ELSE IF(J.GT.JDIM-1) THEN
          JX=J-(JDIM-JWRAP)
      ELSE
          JX=J
      END IF
C
C      COMPUTE DVEFDI
C
      IF(I.EQ.IMIN) THEN
          DVEFDI=(1.-BT+FT)*(-1.5*(V(IMIN,JX,ITT)+AALAM*VM(IMIN,JX,ITT))
1          +2.*(V(IMIN+1,JX,ITT)+AALAM*VM(IMIN+1,JX,ITT))
2          -.5*(V(IMIN+2,JX,ITT)+AALAM*VM(IMIN+2,JX,ITT)))
3          +(BT-FT)*(-1.5*(V(IMIN,JX,ITT+1)+AALAM*VM(IMIN,JX,ITT+1))
4          +2.*(V(IMIN+1,JX,ITT+1)+AALAM*VM(IMIN+1,JX,ITT+1))
5          -.5*(V(IMIN+2,JX,ITT+1)+AALAM*VM(IMIN+2,JX,ITT+1)))
      ELSE IF(I.EQ.IMAX) THEN
          DVEFDI=(1.-BT+FT)*(-1.5*(V(IMAX,JX,ITT)+AALAM*VM(IMAX,JX,ITT))
1          +2.*(V(IMAX-1,JX,ITT)+AALAM*VM(IMAX-1,JX,ITT))
2          -.5*(V(IMAX-2,JX,ITT)+AALAM*VM(IMAX-2,JX,ITT)))
3          +(BT-FT)*(-1.5*(V(IMAX,JX,ITT+1)+AALAM*VM(IMAX,JX,ITT+1))
4          +2.*(V(IMAX-1,JX,ITT+1)+AALAM*VM(IMAX-1,JX,ITT+1))
5          -.5*(V(IMAX-2,JX,ITT+1)+AALAM*VM(IMAX-2,JX,ITT+1)))
      ELSE IF(I.GT.IMIN.AND.I.LT.IMAX) THEN
          DVEFDI=((1.-BT+FT)*(V(I+1,JX,ITT)+AALAM*VM(I+1,JX,ITT))
1          -V(I-1,JX,ITT)-AALAM*VM(I-1,JX,ITT))+
2          (BT-FT)*(V(I+1,JX,ITT+1)+AALAM*VM(I+1,JX,ITT+1))
3          -V(I-1,JX,ITT+1)-AALAM*VM(I-1,JX,ITT+1))/2.
      ELSE
          WRITE (LUERR,*) 'Error in Function DVEFDI, I out of range.'
          WRITE (LUERR,*) 'I = ',I,' JX = ',JX,' ITT = ',ITT
          WRITE (LUERR,*) ' '
      END IF
C
      RETURN
      END
      FUNCTION DVEFDJ(I,J,BT,AALAM,V,VM,IDIM,JDIM,ITMDIM)
C
C      VERSION 1.00
C      1.01A
C      DATE: 12.16.87
C      02.02.89
C
C      PURPOSE: FUNCTION SUBPROGRAM TO COMPUTE THE J DERIVATIVE OF THE
C      EFFECTIVE POTENTIAL AT GRID POINT (I,J) AT NORMALIZED
C      TIME BT.
C
C      INPUT:
C      I          I COORDINATE OF THE GRID POINT
C      J          J COORDINATE OF THE GRID POINT
C      BT         NORMALIZED TIME
C      AALAM      ENERGY INVARIANT (EV (RE/NT)**2/3)
C      V          ELECTRIC POTENTIAL ARRAY (VOLTS)
C      VM         (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3
C      IDIM       I DIMENSION OF V AND VM
C      JDIM       J DIMENSION OF V AND VM
C      ITMDIM     TIME DIMENSION OF V AND VM
C      JWRAP      J WRAP AROUND
```

55

MSM34570
MSM34580
MSM34590
MSM34600
MSM34610
MSM34620
MSM34630
MSM34640
MSM34650
MSM34660
MSM34670
MSM34680
MSM34690
MSM34700
MSM34710
MSM34720
MSM34730
MSM34740
MSM34750
MSM34760
MSM34770
MSM34780
MSM34790
MSM34800
MSM34810
MSM34820
MSM34830
MSM34840
MSM34850
MSM34860
MSM34870
MSM34880
MSM34890
MSM34900
MSM34910
MSM34920
MSM34930
MSM34940
MSM34950
MSM34960
MSM34970
MSM34980
MSM34990
MSM35000
MSM35010
MSM35020
MSM35030
MSM35040
MSM35050
MSM35060
MSM35070
MSM35080
MSM35090
MSM35100
MSM35110
MSM35120
MSM35130
MSM35140
MSM35150
MSM35160
MSM35170
MSM35180
MSM35190
MSM35200

90/06/30
12:03:44

56

msm1.for

```
COMMON /LUNIT/ LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,
1   LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,
2   LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,
3   LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH
DIMENSION A (IMAX, JMAX, KMAX)
DIMENSION NDX (3), NDIM (3), BV (3), COEF (3, 2)

C
C PREPARE INDICES FOR INTERPOLATION
C
NDIM (1) = IMAX
NDIM (2) = JMAX
NDIM (3) = KMAX
BV (1) = BI
BV (2) = BJ
BV (3) = BK
DO 10 L = 1, 3
NDX (L) = BV (L)
IF (NDX (L) .LT. 1) NDX (L) = 1
IF (NDX (L) .GT. NDIM (L) - 1) NDX (L) = NDIM (L) - 1
IF (NDX (L) .LE. 0) NDX (L) = 1
C
FNDX = REAL (NDX (L))
COEF (L, 1) = 1. - BV (L) + FNDX
COEF (L, 2) = BV (L) - FNDX
10 CONTINUE
C
C
G3NTRP = 0.
KSTOP = MIN (KMAX, 2)
JSTOP = MIN (JMAX, 2)
DO 20 I = 1, 2
DO 20 J = 1, JSTOP
DO 20 K = 1, KSTOP
G3NTRP = G3NTRP +
1   COEF (1, I) * COEF (2, J) * COEF (3, K) * A (NDX (1) + I - 1, NDX (2) + J - 1,
2   NDX (3) + K - 1)
20 CONTINUE
RETURN
END
SUBROUTINE MODFIX (BBJ, JDIM, JWRAP)
C
C VERSION 1.00 DATE: 01.15.88
C
C PURPOSE: SUBROUTINE TO ADJUST BBJ TO BE BETWEEN 2 AND JDIM-1
C
C INPUT:
C   BBJ VALUE TO BE ADJUSTED
C   JDIM J DIMENSION OF ARRAYS
C   JWRAP J WRAP AROUND
C
COMMON /LUNIT/ LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,
1   LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,
2   LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,
3   LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH
IF (BBJ .LT. REAL (2)) BBJ = BBJ + REAL (JDIM - JWRAP)
IF (BBJ .GT. REAL (JDIM - 1)) BBJ = BBJ - REAL (JDIM - JWRAP)
RETURN
END
SUBROUTINE MOVER (T, DTRY, BOLD, BJOLD, EPS, BBISCL, BJSCL, DTUSED,
1 DTNEXT, AALAM, V, VM, TIMTAG, ALPHA, BETA, BIR, IDIM, JDIM, ITMDIM, BBI, BBJ)
C
C VERSION 1.00 DATE: 01.15.88
C
C PURPOSE: SUBROUTINE TO ADVANCE ONE PARTICLE LOCATION ONE TIME
```

90/06/30
12:03:44

msm1.for

58

```
C TAKE TWO HALF STEPS
C
      DT2=DT/2.
      CALL RK4 (TSAV,DT2,BBISAV,BBJSAB,BBIOUT,BBJOUT,AALAM,V,VM,
1        TIMTAG,ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM)
C
C SECOND HALF STEP
C
      BBIIN=BBIOUT
      BBJIN=BBJOUT
      CALL RK4 (T+DT2,DT2,BBIIN,BBJIN,BBIA,BBJA,AALAM,V,VM,
1        TIMTAG,ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM)
C
C TAKE LARGE STEP
C
      CALL RK4 (TSAV,DT,BBISAV,BBJSAB,BBIB,BBJB,AALAM,V,VM,
1        TIMTAG,ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM)
C
C EVALUATE ACCURACY
C
      ERRMAX=0.
C
      BIERR=BBIB-BBIA
      ERRMAX=MAX (ERRMAX,ABS (BIERR/BBISCL) )
C
      BJERR=BBJB-BBJA
      ERRMAX=MAX (ERRMAX,ABS (BJERR/BBJSCL) )
C
C SCALE RELATIVE TO REQUIRED TOLERANCE
C
      ERRMAX=ERRMAX/EPS
C
      IF (ERRMAX.GT.ONE) THEN
C          ERROR TOO LARGE, REDUCE STEPSIZE
          DT=SAFETY*DT*(ERRMAX**PSHRNK)
          GO TO 10
C          FOR ANOTHER TRY
      ELSE
          DTUSED=DT
          IF (ERRMAX.GT.ERRCON) THEN
              DTNEXT=SAFETY*DT*(ERRMAX**PGROW)
          ELSE
              DTNEXT=4.*DT
          END IF
      END IF
C
C PUT IN FIFTH ORDER CORRECTION
C
      BBI=BBIA+BIERR*FCOR
      IF (BBI.GT.REAL (IDIM) ) BBI=REAL (IDIM)
      BBJ=BBJA+BJERR*FCOR
C
C
      RETURN
      END
      SUBROUTINE RK4 (T,DT,BBIIN,BBJIN,BBIOUT,BBJOUT,AALAM,V,VM,TIMTAG,
1        ALPHA,BETA,BIR,IDIM,JDIM,ITMDIM)
C
C VERSION 1.00 DATE: 12.30.87
C
C PURPOSE: SUBROUTINE TO ADVANCE ONE PARTICLE LOCATION ONE TIME
C STEP DT USING 4TH ORDER RUNGE KUTTA ALGORITHM.
C
C INPUT:
```

MSM36490
MSM36500
MSM36510
MSM36520
MSM36530
MSM36540
MSM36550
MSM36560
MSM36570
MSM36580
MSM36590
MSM36600
MSM36610
MSM36620
MSM36630
MSM36640
MSM36650
MSM36660
MSM36670
MSM36680
MSM36690
MSM36700
MSM36710
MSM36720
MSM36730
MSM36740
MSM36750
MSM36760
MSM36770
MSM36780
MSM36790
MSM36800
MSM36810
MSM36820
MSM36830
MSM36840
MSM36850
MSM36860
MSM36870
MSM36880
MSM36890
MSM36900
MSM36910
MSM36920
MSM36930
MSM36940
MSM36950
MSM36960
MSM36970
MSM36980
MSM36990
MSM37000
MSM37010
MSM37020
MSM37030
MSM37040
MSM37050
MSM37060
MSM37070
MSM37080
MSM37090
MSM37100
MSM37110
MSM37120

90/06/30
12:03:44

msm1.for

59

```
C      T      TIME PARAMETER      MSM37130
C      DT      TIME STEP, DT.LT.0 MOVES PARTICLES BACK IN TIME      MSM37140
C                      DT.GT.0 MOVES PARTICLES FORWARD      MSM37150
C      BBIIN    CURRENT I LOCATION OF PARTICLE      MSM37160
C      BBJIN    CURRENT J LOCATION OF PARTICLE      MSM37170
C      AALAM    ENERGY INVARIANT OF PARTICLE      MSM37180
C      V        ELECTRIC POTENTIAL ARRAY (VOLTS)      MSM37190
C      VM        (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3      MSM37200
C      TIMTAG    VECTOR GIVING UNIVERSAL TIMES FOR WHICH DATA ARRAYS      MSM37210
C                      ARE GIVEN      MSM37220
C      ALPHA    LATITUDINAL GRID SPACING VECTOR      MSM37230
C      BETA      LOCAL TIME GRID SPACING VECTOR      MSM37240
C      BIR        IONSOPHERIC MAGNETIC FIELD STRENGTH      MSM37250
C      IDIM      NUMBER OF LATITUDINAL GRID POINTS      MSM37260
C      JDIM      NUMBER OF LOCAL TIME GRID POINTS + WRAPAROUND      MSM37270
C      ITMDIM    MAX NUMBER OF TEMPORAL GRID POINTS      MSM37280
C                      MSM37290
C      OUTPUT:      MSM37300
C      BBIOUT    COMPUTED I LOCATION AT END OF TIME STEP      MSM37310
C      BBJOUT    COMPUTED J LOCATION AT END OF TIME STEP      MSM37320
C                      MSM37330
C      PROGRAMMER: R. W. SPIRO      MSM37340
C                      MSM37350
C      REFERENCE:  ABRAMOWITZ AND STEGUN, HANDBOOK OF MATHEMATICAL FUNCTIONS      MSM37360
C                      NATIONAL BUREAU OF STANDARDS, 1970. P. 896, SECT. 25.5.10      MSM37370
C                      MSM37380
C                      MSM37390
C      COMMON /GRID/ DLAM,DPSI,RI,JWRAP      MSM37400
C      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,      MSM37410
C      1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,      MSM37420
C      2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,      MSM37430
C      3      LUCOLT,LUBNDJ,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH      MSM37440
C                      MSM37450
C      DIMENSION ALPHA(IDIM), BETA(IDIM), BIR(IDIM,JDIM)      MSM37460
C      DIMENSION V(IDIM,JDIM,ITMDIM), VM(IDIM,JDIM,ITMDIM), TIMTAG(ITMDIM)      MSM37470
C      DIMENSION BITEMP(4), BJTEMP(4)      MSM37480
C                      MSM37490
C                      MSM37500
C      EVALUATE DERIVATIVE AT TIME=T, LOCATION=(BBI,BBJ)      MSM37510
C      BT=TNORML(T,TIMTAG,ITMDIM)      MSM37520
C      BIUSE=BBIIN      MSM37530
C      BJUSE=BBJIN      MSM37540
C      CALL VLOCTY(BIUSE,BJUSE,BT,AALAM,ALPHA,BETA,BIR,V,VM,IDIM,JDIM,      MSM37550
C      1      ITMDIM,DBIDT,DBJDT)      MSM37560
C      BITEMP(1)=DT*DBIDT      MSM37570
C      BJTEMP(1)=DT*DBJDT      MSM37580
C                      MSM37590
C      EVALUATE DERIVATIVE AT TIME=T+DT/2, LOCATION=(BBIIN+BITEMP(1)/2,      MSM37600
C                      BBJIN+BJTEMP(1)/2)      MSM37610
C      BT=TNORML(T+DT/2.,TIMTAG,ITMDIM)      MSM37620
C      BIUSE=BBIIN+.5*BITEMP(1)      MSM37630
C      BJUSE=BBJIN+.5*BJTEMP(1)      MSM37640
C      CALL VLOCTY(BIUSE,BJUSE,BT,AALAM,ALPHA,BETA,BIR,V,VM,IDIM,JDIM,      MSM37650
C      1      ITMDIM,DBIDT,DBJDT)      MSM37660
C      BITEMP(2)=DT*DBIDT      MSM37670
C      BJTEMP(2)=DT*DBJDT      MSM37680
C                      MSM37690
C      EVALUATE DERIVATIVE AT TIME=T+DT/2, LOCATION=(BBIIN+BITEMP(2)/2,      MSM37700
C                      BBJIN+BJTEMP(2)/2)      MSM37710
C      BIUSE=BBIIN+.5*BITEMP(2)      MSM37720
C      BJUSE=BBJIN+.5*BJTEMP(2)      MSM37730
C      CALL VLOCTY(BIUSE,BJUSE,BT,AALAM,ALPHA,BETA,BIR,V,VM,IDIM,JDIM,      MSM37740
C      1      ITMDIM,DBIDT,DBJDT)      MSM37750
C      BITEMP(3)=DT*DBIDT      MSM37760
```

90/06/30
12:03:44

msm1.for

60

```
C      BJTEMP(3)=DT*DBJDT
C
C      EVALUATE DERIVATIVE AT TIME=T+DT, LOCATION=(BBIIN+BITEMP(3),
C      BBJIN+BJTEMP(3))
      BT=TNORML(T+DT,TIMTAG,ITMDIM)
      BIUSE=BBIIN+BITEMP(3)
      BJUSE=BBJIN+BJTEMP(3)
      CALL VLOCTY(BIUSE,BJUSE,BT,AALAM,ALPHA,BETA,BIR,V,VM,IDIM,JDIM,
1      ITMDIM,DBIDT,DBJDT)
      BITEMP(4)=DT*DBIDT
      BJTEMP(4)=DT*DBJDT
C
C
C      COMPUTE NEW LOCATIONS AT END OF TIME STEP
      BBIOUT=BBIIN+(BITEMP(1)+2.*(BITEMP(2)+BITEMP(3))+BITEMP(4))/6.
      BBJOUT=BBJIN+(BJTEMP(1)+2.*(BJTEMP(2)+BJTEMP(3))+BJTEMP(4))/6.
      RETURN
      END
      SUBROUTINE TCHK(T,TSTOP,DTNOM,BBI,BBJ,IEUSE,ETABEG,IDIM,JDIM,
1      IEDIM,ETANIL,IEXIT)
C
C      VERSION 1.00          DATE: 01.11.88
C
C      PURPOSE:  SUBROUTINE TO CHECK WHETHER TIME BOUNDARY WAS CROSSED THIS
C      TIME STEP.  IS SO, SET IEXIT=1 AND INTERPOLATE TO FIND
C      ETABEG AT LOCATION (BBI,BBJ).
C
C      INPUT:
C      T          TIME
C      TSTOP      TIME AT WHICH TO END PARTICLE TRACEBACK
C      DTNOM      NOMINAL TIME STEP TO BEGIN TRACES.
C      BBI        I LOCATION OF PARTICLE
C      BBJ        J LOCATION OF PARTICLE
C      IEUSE      ENERGY INDEX OF PARTICLS
C      ETABEG     INITIAL ETA DISTRIBUTION (IE., AT T=TSTOP)
C      IDIM       NUMBER OF LATITUDINAL GRID POINTS
C      JDIM       NUMBER OF LOCAL TIME GRID POINTS (INCL WRAPAROUND)
C      IEDIM      NUMBER OF ENERGY SPECIES
C
C      OUTPUT:
C      ETANIL     ETABEG VALUE AT LOCATION (BBI,BBJ) AT TIME T FOR
C      ENERGY SPECIES IEUSE IF PARTICLE HAS CROSSED
C      TIME BOUNDARY
C      IEXIT      SET EQUAL TO 1 IF PARTICLE HAS CROSSED TIME BNDY
C
C      PROGRAMMER: R. W. SPIRO
C
C      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
      DIMENSION ETABEG(IDIM,JDIM,IEDIM)
C
C      IF TIME BOUNDARY IS CROSSED, SET IEXIT=1 AND INTERPOLATE
C      TO FIND ETABEG AT LOCATION (BBI,BBJ)
C
      IF((ABS(T-TSTOP).LT..01*ABS(DTNOM)).OR.
1      (DTNOM.GT.0..AND.T.GE.TSTOP).OR.
2      (DTNOM.LT.0..AND.T.LE.TSTOP)) THEN
```

90/06/30
12:03:44

msml.for

61

```

      IEXIT=1
      RIEUSE=REAL (IEUSE)
      ETANIL=G3NTRP (ETABEG, IDIM, JDIM, IEDIM, BBI, BBJ, RIEUSE)
      END IF
C
      RETURN
      END
      FUNCTION TNORML (TT, TIMTAG, ITMMAX)
C
C      VERSION 1.00
C
C      DATE: 12.16.87
C
C      PURPOSE:  FUNCTION TO CALCULATE RUN-NORMALIZED TIME CORRESPONDING
C                TO TIME TT
C
C      INPUT:
C      TT              TIME IN SECONDS
C      TIMTAG          VECTOR GIVING TIMES FOR WHICH MODEL ARRAYS
C                     ARE CALCULATED
C      ITMMAX          NUMBER OF TEMPORAL GRID POINTS USED
C
C      OUTPUT:
C      TNORML          NON-INTEGGER TIME INDEX CORRESPONDING TO
C                     TIME TT
C
C      PROGRAMMER:  R. W. SPIRO
C
C
C      COMMON /LUNIT/LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,
1      LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIX, LUEAVG, LUFLSM, LUFLX,
2      LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,
3      LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH
      DIMENSION TIMTAG (ITMMAX)
C
      DO 10 I=1, ITMMAX-1
      IF ((TT.GE.TIMTAG(I).AND.TT.LE.TIMTAG(I+1)).OR.
1      (TT.LE.TIMTAG(I).AND.TT.GE.TIMTAG(I+1))) THEN
      II=I
      GO TO 20
      END IF
10 CONTINUE
C
C      TIME TT NOT WITHIN TIMTAG RANGE
C
      WRITE (LUERR, *) 'TT NOT WITHIN TIMTAG RANGE, TT= ', TT
      WRITE (LUERR, *) 'TIMTAG(1)=', TIMTAG(1)
      WRITE (LUERR, *) 'ITMMAX=', ITMMAX
      WRITE (LUERR, *) 'TIMTAG(ITMMAX)=', TIMTAG(ITMMAX)
      WRITE (LUERR, *) 'STOPPING PROGRAM IN FUCTION TNORML'
      STOP
C
C      LOCALIZED TT TO WITHIN THE RIGHT TIME RANGE
C
20 CONTINUE
      TNORML=REAL (II) + (TT-TIMTAG(II)) / (TIMTAG(II+1)-TIMTAG(II))
C
      RETURN
      END
      SUBROUTINE VLOCTY (BBII, BBJJ, BT, AALAM, ALPHA, BETA, BIR, V, VM,
1      IDIM, JDIM, ITMDIM, DBIDT, DBJDT)
C
C      VERSION 1.00
C      1.01A
C
C      DATE: 12.16.87
C      02.02.89
C
```

MSM38410
MSM38420
MSM38430
MSM38440
MSM38450
MSM38460
MSM38470
MSM38480
MSM38490
MSM38500
MSM38510
MSM38520
MSM38530
MSM38540
MSM38550
MSM38560
MSM38570
MSM38580
MSM38590
MSM38600
MSM38610
MSM38620
MSM38630
MSM38640
MSM38650
MSM38660
MSM38670
MSM38680
MSM38690
MSM38700
MSM38710
MSM38720
MSM38730
MSM38740
MSM38750
MSM38760
MSM38770
MSM38780
MSM38790
MSM38800
MSM38810
MSM38820
MSM38830
MSM38840
MSM38850
MSM38860
MSM38870
MSM38880
MSM38890
MSM38900
MSM38910
MSM38920
MSM38930
MSM38940
MSM38950
MSM38960
MSM38970
MSM38980
MSM38990
MSM39000
MSM39010
MSM39020
MSM39030
MSM39040

90/06/30
12:03:44

msml.for

62

```
C  PURPOSE:  SUBROUTINE TO COMPUTE I AND J COMPONENTS OF PARTICLE      MSM39050
C  VELOCITY WITH ENERGY INVARIANT AALAM AT LOCATION                  MSM39060
C  (BBI,BBJ) AT NORMALIZED TIME BT.                                  MSM39070
C                                                                    MSM39080
C  INPUT:                                                                    MSM39090
C  BBI      I LOCATION OF PARTICLE                                      MSM39100
C  BBJ      J LOCATION OF PARTICLE                                      MSM39110
C  BT       NORMALIZED TIME                                           MSM39120
C  AALAM    ENERGY INVARIANT (EV (RE/NT)**2/3)                      MSM39130
C  ALPHA    LATITUDINAL GRID SPACING VECTOR                          MSM39140
C  BETA     LONGITUDINAL GRID SPACING VECTOR                          MSM39150
C  BIR      IONOSPHERIC RADIAL MAGNETIC FIELD STRENGTH              MSM39160
C  V        ELECTRIC POTENTIAL ARRAY (VOLTS)                         MSM39170
C  VM       (FLUX TUBE VOLUME)**-2/3 ARRAY (RE/NT)**-2/3            MSM39180
C  IDIM     1ST (LAT) DIMENSION OF V, VM, AND BIR                   MSM39190
C  JDIM     2ND (LONG) DIMENSION OF V,VM, AND BIR                   MSM39200
C  ITMDIM   3RD (TIME) DIMENSION OF V AND VM                        MSM39210
C                                                                    MSM39220
C  OUTPUT:                                                                    MSM39230
C  DBIDT    I (LAT) VELOCITY OF PARTICLE (I UNITS/SEC)             MSM39240
C  DBJDT    J (LAT) VELOCITY OF PARTICLE (J UNITS/SEC)             MSM39250
C                                                                    MSM39260
C  PROGRAMMER:  R. W  SPIRO                                           MSM39270
C                                                                    MSM39280
C                                                                    MSM39290
C                                                                    MSM39300
C  COMMON /GRID/  DLAM,DPSI,RI,JWRAP                                  MSM39310
C  COMMON /LUNIT/LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,      MSM39320
1  LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,      MSM39330
2  LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,  MSM39340
3  LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH      MSM39350
C                                                                    MSM39360
C  DIMENSION ALPHA(IDIM), BETA(IDIM), BIR(IDIM,JDIM)                MSM39370
C  DIMENSION V(IDIM,JDIM,ITMDIM), VM(IDIM,JDIM,ITMDIM)            MSM39380
C  DIMENSION DVDI2(2,2), DVDJ2(2,2)                                MSM39390
C                                                                    MSM39400
C  BBI=BBI      MSM39410
C  BBJ=BBJ      MSM39420
C  CALL MODFIX(BBJ,JDIM,JWRAP)                                       MSM39430
C  INTERPOLATE GRID ARRAY FOR LOCATION (BBI,BBJ)                    MSM39440
C                                                                    MSM39450
C  BIR2=G3NTRP(BIR,IDIM,JDIM,1,BBI,BBJ,1.)                         MSM39460
C  ALPH2=G3NTRP(ALPHA,IDIM,1,1,BBI,1.,1.)                           MSM39470
C  BETA2=G3NTRP(BETA,IDIM,1,1,BBI,1.,1.)                           MSM39480
C                                                                    MSM39490
C  COMPUTE DVEFDI AND DVEFDJ AT NORMALIZED TIME BT FOR GRID PTS    MSM39500
C  SURROUNDING (BBI,BBJ)                                           MSM39510
C                                                                    MSM39520
C  I=BBI      MSM39530
C  FI=REAL(I)  MSM39540
C  J=BBJ      MSM39550
C  FJ=REAL(J)  MSM39560
C                                                                    MSM39570
C                                                                    MSM39580
C  DO 10 II=1,2  MSM39590
C    IUSE=I+II-1  MSM39600
C    IF(IUSE.GT.IDIM) IUSE=IDIM  MSM39610
C  DO 20 JJ=1,2  MSM39620
C    JUSE=J+JJ-1  MSM39630
C    DVDI2(II,JJ)=DVEFDI(IUSE,JUSE,BT,AALAM,V,VM,IDIM,JDIM,ITMDIM)  MSM39640
C    DVDJ2(II,JJ)=DVEFDJ(IUSE,JUSE,BT,AALAM,V,VM,IDIM,JDIM,ITMDIM)  MSM39650
20 CONTINUE  MSM39660
10 CONTINUE  MSM39670
C  INTERPOLATE DVDI2 AND DVDJ2 FOR LOCATION (BBI,BBJ)              MSM39680
```

90/06/30
12:03:44

msml.for

63

```
C
  DVDI=(1.-BBJ+FJ)*(1.-BBI+FI)*DVDI2(1,1)+(BBI-FI)*DVDI2(2,1)
1    + (BBJ-FJ)*(1.-BBI+FI)*DVDI2(1,2)+(BBI-FI)*DVDI2(2,2)
C
  DVDJ=(1.-BBJ+FJ)*(1.-BBI+FI)*DVDJ2(1,1)+(BBI-FI)*DVDJ2(2,1)
1    + (BBJ-FJ)*(1.-BBI+FI)*DVDJ2(1,2)+(BBI-FI)*DVDJ2(2,2)
C
C  COMPUTE VELOCITIES
C
  DBIDT=DVDJ*1000./(BIR2*ALPH2*BETA2*DLAM*DPSI*RI**2)
  DBJDT=-DVDI*1000./(BIR2*ALPH2*BETA2*DLAM*DPSI*RI**2)
C
  RETURN
  END
  SUBROUTINE RDGRID(IDIM,JDIM,ALPHA,BETA,TETA,PHI,COLAT,BIR,SINI)
C
C  VERSION 1.0                      DATE: 01.23.88
C
C  PROGRAMMER: R. W. SPIRO
C
C  PURPOSE:  READ GRID SYSTEM COORDINATES AND CALCULATE ESSENTIAL GRID
C            QUANTITIES
C
  COMMON /GRID/ DLAM,DPSI,RI,JWRAP
  COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
1    LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
2    LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
3    LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
  DIMENSION      ALPHA(IDIM),BETA(IDIM),TETA(IDIM),PHI(JDIM)
  DIMENSION COLAT(IDIM,JDIM),BIR(IDIM,JDIM),SINI(IDIM,JDIM)
C
  CHARACTER      ID(2)
  PI=ATAN2(0.,-1.)
  JWRAP = 3
C
C  READ GRID PARAMETERS
C
  OPEN(UNIT=LUCORD,FILE='COORD',STATUS='OLD')
C
  READ(LUCORD,1000) ID,OFFSET,DLAM,DPSI,RI,RE
1000  FORMAT(2A4,1X,5(E12.6,1X))
  READ(LUCORD,1001)TETA,ALPHA,BETA,PHI
1001  FORMAT(10(E12.6,1X))
C
  DO 10 I=1,IDIM
  DO 20 J=1,JDIM
    COSTET=COS(TETA(I))*COS(OFFSET)+SIN(TETA(I))*COS(PHI(J))*
1    SIN(OFFSET)
    BIR(I,J)=62000.*COSTET
    COLAT(I,J)=ACOS(COSTET)
    SINI(I,J)=2.*COSTET/SQRT(1.+3.*COSTET**2)
20  CONTINUE
10  CONTINUE
C
  RETURN
  END
  SUBROUTINE MLTSET(ISEC,IDIM,JDIM,ALOCT)
C
C  VERSION 1.0                      DATE: 01.23.88
C
C  PROGRAMMER: R. W. SPIRO
C
C  PURPOSE:  CALCULATE TIME DEPENDENT ALOCT OF ROTATING GRID PTS
```

MSM39690
MSM39700
MSM39710
MSM39720
MSM39730
MSM39740
MSM39750
MSM39760
MSM39770
MSM39780
MSM39790
MSM39800
MSM39810
MSM39820
MSM39830
MSM39840
MSM39850
MSM39860
MSM39870
MSM39880
MSM39890
MSM39900
MSM39910
MSM39920
MSM39930
MSM39940
MSM39950
MSM39960
MSM39970
MSM39980
MSM39990
MSM40000
MSM40010
MSM40020
MSM40030
MSM40040
MSM40050
MSM40060
MSM40070
MSM40080
MSM40090
MSM40100
MSM40110
MSM40120
MSM40130
MSM40140
MSM40150
MSM40160
MSM40170
MSM40180
MSM40190
MSM40200
MSM40210
MSM40220
MSM40230
MSM40240
MSM40250
MSM40260
MSM40270
MSM40280
MSM40290
MSM40300
MSM40310
MSM40320

90/06/30
12:01:44

msml.for

64

```
C
C INPUTS:
C   ISEC          TIME (SECONDS)
C   IDIM          I DIMENSION
C   JDIM          J DIMENSION
C
C OUTPUT:
C   ALOCT          LOCAL TIME ARRAY CORRESPONDING TO T=ISEC
C
C   COMMON /GRID/ DLAM,DPSI,RI,JWRAP
C   DIMENSION ALOCT(IDIM,JDIM)
C
C   PI=ATAN2(0.,-1.)
C   DPHI=2.*PI/REAL(JDIM-JWRAP)
C   DPHIDT=2.*PI/86400.
C   T=REAL(ISEC)
C
C   DO 10 I=1,IDIM
C   DO 10 J=1,JDIM
C       ALOCT(I,J)=REAL(J-3)*DPHI+T*DPHIDT
C       IF (ALOCT(I,J).GT.2.*PI) ALOCT(I,J)=ALOCT(I,J)-2.*PI
C       IF (ALOCT(I,J).LT.0.) ALOCT(I,J)=ALOCT(I,J)+2.*PI
C 10 CONTINUE
C
C   RETURN
C   END
C   SUBROUTINE VMLSET(ITMCUR,VM,BIR,SINI,IDIM,JDIM,ITMDIM,VMLOSS)
C
C   VERSION 1.0                      DATE: 01.23.88
C
C   PROGRAMMER: R. W. SPIRO
C
C   PURPOSE:  COMPUTE VMLOSS ARRAY FOR ITMCUR
C
C   DIMENSION VMLOSS(IDIM,JDIM,ITMDIM),BIR(IDIM,JDIM),SINI(IDIM,JDIM)
C   DIMENSION VM(IDIM,JDIM,ITMDIM)
C
C   DO 10 I=1,IDIM
C   DO 10 J=1,JDIM
C       VMLOSS(I,J,ITMCUR)=(SINI(I,J)/BIR(I,J))*VM(I,J,ITMCUR)**2
C 10 CONTINUE
C
C   RETURN
C   END
C   SUBROUTINE FNDBND(ITMCUR,COLAT,ALOCT,A2,B2,DX2,DY2,IDIM,JDIM,
C   1          ITMDIM,BNDLOC)
C
C   VERSION 1.0                      DATE: 01.23.88
C
C   PROGRAMMER: R. W. SPIRO
C
C   PURPOSE:  COMPUTE TIME DEPENDENT OUTER BOUNDARY LOCATION
C
C   COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
C   1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
C   2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
C   3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
C   DIMENSION COLAT(IDIM,JDIM),ALOCT(IDIM,JDIM)
C   DIMENSION BNDLOC(JDIM,ITMDIM)
C
C   PI = ATAN2(0.,-1.)
C   DO 10 J=1,JDIM
C   DO 20 I=1,IDIM-1
```

MSM40330
MSM40340
MSM40350
MSM40360
MSM40370
MSM40380
MSM40390
MSM40400
MSM40410
MSM40420
MSM40430
MSM40440
MSM40450
MSM40460
MSM40470
MSM40480
MSM40490
MSM40500
MSM40510
MSM40520
MSM40530
MSM40540
MSM40550
MSM40560
MSM40570
MSM40580
MSM40590
MSM40600
MSM40610
MSM40620
MSM40630
MSM40640
MSM40650
MSM40660
MSM40670
MSM40680
MSM40690
MSM40700
MSM40710
MSM40720
MSM40730
MSM40740
MSM40750
MSM40760
MSM40770
MSM40780
MSM40790
MSM40800
MSM40810
MSM40820
MSM40830
MSM40840
MSM40850
MSM40860
MSM40870
MSM40880
MSM40890
MSM40900
MSM40910
MSM40920
MSM40930
MSM40940
MSM40950
MSM40960

90/06/30
12:03:44

msml.for

65

```
IUSE=I
THETAB=THET(A2,B2,DX2,DY2,ALOC(T(I+1,J))*PI/180.
IF(COLAT(I+1,J).GE.THETAB) GO TO 30
20 CONTINUE
C
C ERROR EXIT
WRITE(LUERR,*) 'ERROR IN FFNDBND'
WRITE(LUERR,*) 'THETAB=',THETAB,'J=',J,'I=',I
WRITE(LUERR,*) 'STOPPING PGM'
STOP
C
30 CONTINUE
C
THETAC=THET(A2,B2,DX2,DY2,ALOC(T(IUSE,J))*PI/180.
BNDLOC(J,ITMCUR)=(REAL(IUSE)*(COLAT(IUSE+1,J)-THETAB)+
1 REAL(IUSE+1)*(THETAC-COLAT(IUSE,J)))/
2 (COLAT(IUSE+1,J)-COLAT(IUSE,J)-THETAB+THETAC)
C
10 CONTINUE
C
RETURN
END
SUBROUTINE OUTP(R,ISIZE,JSIZE,IBEG,IEND,IINC,JBEG,JEND,JINC,
$ XSCALE,TITLE,NTP,NCOL)
C
C AUTHOR: R.W. SPIRO LAST MODIFIED: 06-25-85
C
C*****
C DESCRIPTION OF INPUT PARAMETERS
C*****
C R(ISIZE,JSIZE)=ARRAY TO BE OUTPUT
C THIS SUBROUTINE IS ABLE TO OUTPUT SELECTED ELEMENTS OF ARRAY
C IBEG= INITIAL I VALUE TO BE OUTPUT
C IEND= FINAL I VALUE TO BE OUTPUT
C IINC= I VALUE INCREMENT FOR OUTPUT
C (NOTE: (IEND-IBEG)/IINC SHOULD BE AN INTEGER)
C JBEG,JEND, AND JINC ARE DEFINED SIMILARLY
C SCALE=SCALE FACTOR
C IF(SCALE.EQ.0.) SCALE IS CALCULATED TO GIVE BEST DISPLAY
C ALL ELEMENTS OR ARRAY ARE DIVIDED BY SCALE BEFORE BEING OUTPUT
C TITLE=CHARACTER STRING THAT IDENTIFIES THE ARRAY BEING OUTPUT
C NTP= OUTPUT UNIT NUMBER
C NCOL= NUMBER OF COLUMNS FOR OUTPUT DEVICE (80 OR 132)
C*****
C
PARAMETER (JCOL=16)
DIMENSION R(ISIZE,JSIZE), Y(JCOL)
REAL*8 SUM0, SUM1, SUM2, TERM, AVE, SD
CHARACTER TITLE *(*)
C
C INITIALIZATION
C
SCALE=XSCALE
ISCLMX=-12
SUM0=0.0
SUM1=0.0
SUM2=0.0
MXLINE=57
JJCOL=(NCOL-4)/8
C
C CALCULATE SIZE OF PRINTOUT ARRAY (NI BY NJ)
C
```

MSM40970
MSM40980
MSM40990
MSM41000
MSM41010
MSM41020
MSM41030
MSM41040
MSM41050
MSM41060
MSM41070
MSM41080
MSM41090
MSM41100
MSM41110
MSM41120
MSM41130
MSM41140
MSM41150
MSM41160
MSM41170
MSM41180
MSM41190
MSM41200
MSM41210
MSM41220
MSM41230
MSM41240
MSM41250
MSM41260
MSM41270
MSM41280
MSM41290
MSM41300
MSM41310
MSM41320
MSM41330
MSM41340
MSM41350
MSM41360
MSM41370
MSM41380
MSM41390
MSM41400
MSM41410
MSM41420
MSM41430
MSM41440
MSM41445
MSM41450
MSM41460
MSM41470
MSM41480
MSM41490
MSM41500
MSM41510
MSM41520
MSM41530
MSM41540
MSM41550
MSM41560
MSM41570
MSM41580
MSM41590

90/06/30
12:03:44

msm1.for

66

```

      NI=(IEND-IBEG)/IINC+1
      NJ=(JEND-JBEG)/JINC+1
C
C  IF SCALE=0. THEN COMPUTE AUTO SCALE FACTOR
C
      IF(SCALE.EQ.0.) THEN
C          DETERMINE MAXIMUM # OF PLACES TO LEFT OF DECIMAL PT
          DO 10 I=IBEG,IEND,IINC
          DO 10 J=JBEG,JEND,JINC
              IF(R(I,J).NE.0..AND.ABS(R(I,J)).GT.1.E-36) THEN
                  TEST=LOG10(ABS(R(I,J)))
                  IF(TEST.GE.0.) THEN
                      TEST=TEST+.000001
                  ELSE
                      TEST=TEST-.999999
                  END IF

                  ITEST=TEST
              ELSE
                  ITEST=0
              END IF
              IF(R(I,J).LT.0.) ITEST=ITEST+1
              ISCLMX=MAX0(ISCLMX,ITEST)
10      CONTINUE
C          DETERMINE SCALE FACTOR SUCH THAT MAX # OF PLACES TO LEFT OF
C          DECIMAL POINT IS 3.
          IPOWER=ISCLMX-2
          SCALE=10.**IPOWER
      ELSE
          IPOWER=NINT(LOG10(SCALE))
      END IF
C
C      COMPUTE MEAN AND STD DEVIATION OF OUTPUTTED ARRAY ELEMENTS
C
C
C      DO 20 I=1,ISIZE
C      DO 20 J=1,JSIZE
          TERM=R(I,J)
          SUM0=SUM0+1.
          IF (ABS(TERM).LT.1.E-20) GO TO 20
          SUM1=SUM1+TERM
          SUM2=SUM2+TERM**2
20      CONTINUE
          AVE=SUM1/SUM0
          SD=SQRT((SUM2-SUM0*AVE**2)/(SUM0-1.))
C
C      OUTPUT DATA
C
C
C          JSTART=JBEG
C          ISTART=IBEG
C          IFINAL=IEND
C          IUTT=0
C          WRITE(NTP,810) TITLE,IPOWER,IUTT,AVE,SD
810      FORMAT('0',4X,A,'/(1.E',I2,')',4X,'UT=',I6,2X,'AVE=',1PE15.8,2X,
$           'SD=',E15.8)
          LCOUNT=0
C
C      BEGIN OUTPUT LOOP
C
C
```

MSM41600
MSM41610
MSM41620
MSM41630
MSM41640
MSM41650
MSM41660
MSM41670
MSM41680
MSM41690
MSM41700
MSM41710
MSM41720
MSM41730
MSM41740
MSM41750
MSM41760
MSM41770
MSM41780
MSM41790
MSM41800
MSM41810
MSM41820
MSM41830
MSM41840
MSM41850
MSM41860
MSM41870
MSM41880
MSM41890
MSM41900
MSM41910
MSM41920
MSM41930
MSM41940
MSM41950
MSM41960
MSM41970
MSM41980
MSM41990
MSM42000
MSM42010
MSM42020
MSM42030
MSM42040
MSM42050
MSM42060
MSM42070
MSM42080
MSM42090
MSM42100
MSM42110
MSM42120
MSM42130
MSM42140
MSM42150
MSM42160
MSM42170
MSM42180
MSM42190
MSM42200
MSM42210
MSM42220
MSM42230

90/06/30
12:03:44

msml.for

67

```
30 CONTINUE
  JFINAL=(JJCOL-1)*JINC+JSTART
  IF(JFINAL.GT.JEND) JFINAL=JEND
C
C FOLLOWING IS MAIN IF STATEMENT IN LOOP
C
  IF(JSTART.LE.JEND) THEN
C
C      OUTPUT A BLOCK
C
  IF(MXLINE-LCOUNT.LE.5) THEN
C
C      START A NEW PAGE
C
    IUTT=0
    WRITE(NTP,810) TITLE,IPOWER,IUTT,AVE,SD
    LCOUNT=0
    END IF
    WRITE(NTP,830) (JJ,JJ=JSTART,JFINAL,JINC)
830 FORMAT('0',3X,16(5X,I3))
    LCOUNT=LCOUNT+2
    DO 40 II=ISTART,IFINAL,IINC
      ICOUNT=LCOUNT+1
      IF(LCOUNT.GT.MXLINE) THEN
C
C          START A NEW PAGE
C
        IUTT=0
        WRITE(NTP,810) TITLE,IPOWER,IUTT,AVE,SD
        WRITE(NTP,830) (JJ,JJ=JSTART,JFINAL,JINC)
        LCOUNT=2
      END IF
      JCOUNT=0
      DO 50 JJ=JSTART,JFINAL,JINC
        JCOUNT=JCOUNT+1
        Y(JCOUNT)=R(II,JJ)/SCALE
50      CONTINUE
        WRITE(NTP,840) II,(Y(JJ),JJ=1,JCOUNT)
840 FORMAT(1X,I3,16(F8.3))
40      CONTINUE
        JSTART=JFINAL+JINC
        ISTART=IBEG
        GO TO 30
      END IF
    RETURN
  END
  SUBROUTINE HIEPAR
C
C PURPOSE: SUBROUTINE TO COMPUTE EMPIRICAL MODEL OF MEV ENERGY
C          ELECTRONS. THIS ROUTINE IS A DUMMY TO REPRESENT THE
C          TRUE SUBROUTINE TO BE FURNISHED BY D. BAKER OF GSFC.
C
  RETURN
END
C
C SUBROUTINE OUTPUT
C
C PURPOSE: DUMMY SUBROUTINE TO REPRESENT ALGORITHM TO OUTPUT MSM
C          RESULTS TO ENVIRONMENTAL DATA BASE. THIS ROUTINE WILL BE
C          THE OUTPUT INTERFACE BETWEEN THE MSM AND THE ENVIRONMENTAL
C          DATA BASE.
C
  RETURN
END
SUBROUTINE AURL1(FKP,COLAT,ALOCT,LATDIM,LTDIM,AEFLUX,AEMEAN)
C
C Determines the number and energy fluxes given Kp, corrected
```

MSM42240
MSM42250
MSM42260
MSM42270
MSM42280
MSM42290
MSM42300
MSM42310
MSM42320
MSM42330
MSM42340
MSM42350
MSM42360
MSM42370
MSM42380
MSM42390
MSM42400
MSM42410
MSM42420
MSM42430
MSM42440
MSM42450
MSM42460
MSM42470
MSM42480
MSM42490
MSM42500
MSM42510
MSM42520
MSM42530
MSM42540
MSM42550
MSM42560
MSM42570
MSM42580
MSM42590
MSM42600
MSM42610
MSM42620
MSM42630
MSM42640
MSM42650
MSM42660
MSM42670
MSM42680
MSM42690
MSM42700
MSM42710
MSM42720
MSM42730
MSM42740
MSM42750
MSM42760
MSM42770
MSM42780
MSM42790
MSM42800
MSM42810
MSM42820
MSM42830
MSM42840
MSM42850
MSM42860
MSM42870

90/06/30
12:03:44

68

msml.for

```
C      geomagnetic latitude, and local time                                MSM42880
C                                                                                               MSM42890
C      FKP=The most recent Kp value                                         MSM42900
C      COLAT=An array listing pertinent colatitudes (less than 40 deg.)    MSM42910
C      preferably using corrected geomagnetic coordinates                  MSM42920
C      ALOCT=An array listing pertinent magnetic local times                MSM42930
C      LATDIM=DIMENSION IN LATITUDE                                         MSM42940
C      LTDIM=DIMENSION IN LOCAL TIME                                        MSM42950
C      AEFLUX=The integral electron energy flux given in units of ergs     MSM42960
C      per cm**2-sec for a given latitude and local time                  MSM42970
C      AEMEAN=AVERAGE ENERGY IN EV                                        MSM42980
C      AFLUX=The integral electron number flux given in units of inverse   MSM42990
C      cm**2-sec for a given latitude and local time                      MSM43000
C                                                                                               MSM43010
C      PARAMETER(IIDIM=62, JJDIM=51)                                         MSM43020
C      COMMON /LUNIT/LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,         MSM43030
1      LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,         MSM43040
2      LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,     MSM43050
3      LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH         MSM43060
C                                                                                               MSM43070
C      DIMENSION ALOCT(LATDIM, LTDIM), COLAT(LATDIM, LTDIM),              MSM43080
1      AEFLUX(LATDIM, LTDIM), AFLUX(IIDIM, JJDIM), AEMEAN(LATDIM, LTDIM), MSM43090
2      RDC(2), H0DC(2), S1DC(2), S2DC(2), RCOS(6,2), RSIN(6,2),           MSM43100
3      H0COS(6,2), H0SIN(6,2), S1COS(6,2), S1SIN(6,2), S2COS(6,2),       MSM43110
4      S2SIN(6,2)                                                         MSM43120
C                                                                                               MSM43130
C      PI=ACOS(-1.)                                                         MSM43140
C                                                                                               MSM43150
C      Read off the model coefficients for the given Kp                    MSM43160
C      CALL FNDHDY(FKP, RDC, RCOS, RSIN, H0DC, H0COS, H0SIN,              MSM43170
1      S1DC, S1COS, S1SIN, S2DC, S2COS, S2SIN, 6,2)                      MSM43180
C                                                                                               MSM43190
C      DO 20 J=1, LTDIM                                                     MSM43200
C      DO 10 I=1, LATDIM                                                    MSM43210
C                                                                                               MSM43220
C      XLT=ALOCT(I, J)                                                      MSM43230
C                                                                                               MSM43240
C                                                                                               MSM43250
C      Determine the values of the energy flux array                       MSM43260
C      CALL ADDHDY(RDC, RCOS, RSIN, H0DC, H0COS, H0SIN, S1DC, S1COS,      MSM43270
1      S1SIN, S2DC, S2COS, S2SIN, 6,2,1, XLT, COLAT(I, J),              MSM43280
2      AEFLUX(I, J))                                                       MSM43290
C                                                                                               MSM43300
C      Convert units from keV/cm**2-ster-sec to erg/cm**2-sec             MSM43310
C      AEFLUX(I, J)=(10.**AEFLUX(I, J))*PI*1.6021E-9                      MSM43320
C                                                                                               MSM43330
C                                                                                               MSM43340
C                                                                                               MSM43350
10      CONTINUE                                                            MSM43360
20      CONTINUE                                                            MSM43370
C                                                                                               MSM43380
C      DO 40 J=1, LTDIM                                                     MSM43390
C      DO 30 I=1, LATDIM                                                    MSM43400
C                                                                                               MSM43410
C      XLT=ALOCT(I, J)                                                      MSM43420
C                                                                                               MSM43430
C      Determine the values of the number flux array                      MSM43440
C      CALL ADDHDY(RDC, RCOS, RSIN, H0DC, H0COS, H0SIN, S1DC, S1COS,      MSM43450
1      S1SIN, S2DC, S2COS, S2SIN, 6,2,2, XLT, COLAT(I, J),              MSM43460
2      AFLUX(I, J))                                                         MSM43470
C                                                                                               MSM43480
C      Convert units from (ster-cm**2-sec)**-1 to (cm**2-sec)**-1         MSM43490
C      AFLUX(I, J)=(10.**AFLUX(I, J))*PI                                   MSM43500
C      IF (AFLUX(I, J).NE.0.0) AEMEAN(I, J)=AEFLUX(I, J)/AFLUX(I, J)     MSM43510
```

90/06/30
12:03:44

msml.for

69

```

      AEMEAN(I,J)=AEMEAN(I,J)*6.24E11
30    CONTINUE
40    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE FNDHDY(FKP,RDC,RCOS,RSIN,H0DC,H0COS,H0SIN,
1  S1DC,S1COS,S1SIN,S2DC,S2COS,S2SIN,MDIM,NDIM)
C      Find and read the pertinent Fourier coefficients
C
      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
1  LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
2  LUEFLX,LUEFLD,LUBF D,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
3  LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
      DIMENSION RDC(NDIM),H0DC(NDIM),S1DC(NDIM),S2DC(NDIM),
1  RCOS(MDIM,NDIM),RSIN(MDIM,NDIM),H0COS(MDIM,NDIM),
2  H0SIN(MDIM,NDIM),S1COS(MDIM,NDIM),S1SIN(MDIM,NDIM),
3  S2COS(MDIM,NDIM),S2SIN(MDIM,NDIM)
C
      DATA IKPCHK /-1/
C
      IFKP2=NINT(FKP)
      IF(IFKP2.GT.6) IFKP2=6
C
C  IF THESE KP COEFS WERE THE LAST READ, DON'T REREAD
C  IF(IKPCHK.EQ.IFKP2) GO TO 99
C
      OPEN(UNIT=LUHDYE,FILE='HARDY',STATUS='OLD')
C
      IF(IREWCK.NE.1)THEN
        GO TO 10
      ELSE IF (IKPCHK.NE.IFKP2)THEN
        REWIND LUHDYE
      ELSE
        GO TO 99
      ENDIF
C
10    IF(IFKP2.EQ.0)GO TO 30
C    Read off irrelevant entries
      DO 20 I=1,IFKP2*26
        READ(LUHDYE,*)COEFF
20    CONTINUE
C
30    DO 60 N=1,NDIM
C
      Read off the constant and trigonometric coefficients
      for the Epstein-Fourier equation
      READ(LUHDYE,*)RDC(N),H0DC(N),S1DC(N),S2DC(N)
C
      DO 40 M=1,MDIM
        READ(LUHDYE,*)RCOS(M,N),H0COS(M,N),S1COS(M,N),S2COS(M,N)
40    CONTINUE
C
      DO 50 M=1,MDIM
        READ(LUHDYE,*)RSIN(M,N),H0SIN(M,N),S1SIN(M,N),S2SIN(M,N)
50    CONTINUE
C
60    CONTINUE
C
      IKPCHK=IFKP2
      IREWCK=1
```

MSM43520
MSM43530
MSM43540
MSM43550
MSM43560
MSM43570
MSM43580
MSM43590
MSM43600
MSM43610
MSM43620
MSM43630
MSM43640
MSM43650
MSM43660
MSM43670
MSM43680
MSM43690
MSM43700
MSM43710
MSM43720
MSM43730
MSM43740
MSM43750
MSM43760
MSM43770
MSM43780
MSM43790
MSM43800
MSM43810
MSM43820
MSM43830
MSM43840
MSM43850
MSM43860
MSM43870
MSM43880
MSM43890
MSM43900
MSM43910
MSM43920
MSM43930
MSM43940
MSM43950
MSM43960
MSM43970
MSM43980
MSM43990
MSM44000
MSM44010
MSM44020
MSM44030
MSM44040
MSM44050
MSM44060
MSM44070
MSM44080
MSM44090
MSM44100
MSM44110
MSM44120
MSM44130
MSM44140
MSM44150

90/06/30
12:03:44

msml.for

70

```
C      CLOSE (LUHDYE)
C
C      99  RETURN
C      END
C
C      SUBROUTINE ADDHDY (RDC, RCOS, RSIN, H0DC, H0COS, H0SIN, S1DC,
1  S1COS, S1SIN, S2DC, S2COS, S2SIN, MDIM, NDIM, K, ALTIME, COLTDE, FLUX)
C      Calculate the Hardy average value for the given point
C
C      DIMENSION RDC (NDIM), H0DC (NDIM), S1DC (NDIM), S2DC (NDIM),
1  RCOS (MDIM, NDIM), RSIN (MDIM, NDIM), H0COS (MDIM, NDIM),
2  H0SIN (MDIM, NDIM), S1COS (MDIM, NDIM), S1SIN (MDIM, NDIM),
3  S2COS (MDIM, NDIM), S2SIN (MDIM, NDIM)
C
C      X=90.-ABS (COLTDE)*180./3.14159
C      ALCT=12.+ALTIME*12./3.14159
C      IF (ALCT.GT.24.) ALCT=ALCT-24.
C
C      Check to see if the Fourier series have to be recalculated
C      IF (TEST.EQ.ALCT) GO TO 10
C
C      Determine the coefficients for the Epstein equation
C      R=FSUM (RDC, RSIN, RCOS, ALCT, 6, 2, K)
C      H0=FSUM (H0DC, H0SIN, H0COS, ALCT, 6, 2, K)
C      S1=FSUM (S1DC, S1SIN, S1COS, ALCT, 6, 2, K)
C      S2=FSUM (S2DC, S2SIN, S2COS, ALCT, 6, 2, K)
C
C      Use the Epstein equation to calculate the average value
10  FLUX=R+S1*(X-H0)+(S2-S1)*ALOG((1-S1/S2*EXP(X-H0))/(1-S1/S2))
C      Set the equatorward lower limit flux value at 6 and
C      the poleward lower limit at 7.
C      IF ((FLUX.LT.6.).AND.(X.LT.70.)) FLUX=6.
C      IF ((FLUX.LT.7.).AND.(X.GT.70.)) FLUX=7.
C
C      TEST=ALCT
C
C      RETURN
C      END
C
C      REAL FUNCTION FSUM (DC, BSIN, BCOS, ALCT, MDIM, NDIM, K)
C      Determines the sum of a Fourier series
C
C      DIMENSION BSIN (MDIM, NDIM), BCOS (MDIM, NDIM), DC (NDIM)
C
C      FSUM=0.
C
C      DO 10 L=1, MDIM
C          FSUM=FSUM+BSIN (L, K)*SIN (L*3.141592*ALCT/12.)+
1  BCOS (L, K)*COS (L*3.141592*ALCT/12.)
10  CONTINUE
C
C      FSUM=FSUM+DC (K)
C
C      RETURN
C      END
C      SUBROUTINE WRT3D (LUN, IRECMX, ID, RID, CHID, ARRAY, IDIM, JDIM, KDIM,
1  ITMDIM)
C
C      PROGRAMMER: BOB SPIRO
C
C      PURPOSE: THIS SUBROUTINE WRITES 3-D ARRAYS INTO THE STANDARD
```

MSM44160
MSM44170
MSM44180
MSM44190
MSM44200
MSM44210
MSM44220
MSM44230
MSM44240
MSM44250
MSM44260
MSM44270
MSM44280
MSM44290
MSM44300
MSM44310
MSM44320
MSM44330
MSM44340
MSM44350
MSM44360
MSM44370
MSM44380
MSM44390
MSM44400
MSM44410
MSM44420
MSM44430
MSM44440
MSM44450
MSM44460
MSM44470
MSM44480
MSM44490
MSM44500
MSM44510
MSM44520
MSM44530
MSM44540
MSM44550
MSM44560
MSM44570
MSM44580
MSM44590
MSM44600
MSM44610
MSM44620
MSM44630
MSM44640
MSM44650
MSM44660
MSM44670
MSM44680
MSM44690
MSM44700
MSM44710
MSM44720
MSM44730
MSM44740
MSM44750
MSM44760
MSM44770
MSM44780
MSM44790

90/06/30
12:03:44

msm1.for

71

```
C      MAGNETOSPHERIC SPECIFICATIONS MODEL FILE FORMAT.
C
      CHARACTER*80 CHID
      DIMENSION ID(20),RID(20),ARRAY(IDIM,JDIM,KDIM)
C
      IRECL =IDIM*JDIM + 2*80
      OPEN (UNIT=LUN,ACCESS='DIRECT',RECL=IRECL,STATUS='UNKNOWN')
C
      ID(12)=KDIM
C
C      WRITE THE ARRAY
C
      DO 10, K=1, KDIM
C
      CALCULATE THE RECORD NUMBER
C
      ID(13)=K
      IREC = (IRECMX-1)*KDIM + K
      WRITE (LUN,REC=IREC) ID,RID,CHID,
2      ((ARRAY(I,J,K),I=1,IDIM),J=1,JDIM)
10  CONTINUE
C
      CLOSE (LUN)
C
      RETURN
END
C
      SUBROUTINE RDHDR(LUN,FILNAM,IRECMX,ID,RID,CHID,IDIM,JDIM,ITMDIM)
C
C      PROGRAMMER: BOB SPIRO          DATE: 6/26/88
C
C      PURPOSE: THIS ROUTINE READS THE HEADER OF THE STANDARD
C      MAGNETOSPHERIC SPECIFICATIONS MODEL FILE FORMAT.
C
      CHARACTER CHID*80, FILNAM*6
      DIMENSION ID(20),RID(20)
C
      OPEN THE FILE
C
      IRECL = IDIM*JDIM + 2*80
      OPEN (UNIT=LUN,ACCESS='DIRECT',RECL=IRECL,STATUS='OLD')
C
      READ THE HEADER
C
      READ (LUN,REC=IRECMX) ID,RID,CHID
C
      CLOSE (LUN)
C
      RETURN
END
C
      SUBROUTINE READ3D(LUN,FILNAM,LREC,IDIM,JDIM,KDIM,ITMDIM,ID,RID,
2      CHID,ARRAY)
C
C      PROGRAMMER: BOB SPIRO          DATE: 6/26/88
C
C      PURPOSE: THIS SUBROUTINE READS A LOGICAL RECORD FROM THE
C      STANDARD MAGNETOSPHERIC SPECIFICATIONS MODEL FILE FORMAT.
C
      CHARACTER CHID*80, FILNAM*6
C
      DIMENSION ARRAY(IDIM,JDIM,KDIM)
      DIMENSION ID(20),RID(20)
```

MSM44800
MSM44810
MSM44820
MSM44830
MSM44840
MSM44850
MSM44860
MSM44870
MSM44880
MSM44890
MSM44900
MSM44910
MSM44920
MSM44930
MSM44940
MSM44950
MSM44960
MSM44970
MSM44980
MSM44990
MSM45000
MSM45010
MSM45020
MSM45030
MSM45040
MSM45050
MSM45060
MSM45070
MSM45080
MSM45090
MSM45100
MSM45110
MSM45120
MSM45130
MSM45140
MSM45150
MSM45160
MSM45170
MSM45180
MSM45190
MSM45200
MSM45210
MSM45220
MSM45230
MSM45240
MSM45250
MSM45260
MSM45270
MSM45280
MSM45290
MSM45300
MSM45310
MSM45320
MSM45330
MSM45340
MSM45350
MSM45360
MSM45370
MSM45380
MSM45390
MSM45400
MSM45410
MSM45420
MSM45430

90/06/30
12:03:44

msml.for

72

```
C
C      OPEN THE FILE
C
C      IRECL = IDIM*JDIM + 2*80
C      OPEN (UNIT=LUN,ACCESS='DIRECT',RECL=IRECL,STATUS='OLD')
C
C      READ THE ARRAY
C
C      DO 10, K=1, KDIM
C
C          CALCULATE THE RECORD NUMBER
C
C          IREC = (LREC-1)*KDIM + K
C          READ (LUN,REC=IREC) ID,RID,CHID,
10      2      ((ARRAY(I,J,K),I=1,IDIM),J=1,JDIM)
C      CONTINUE
C
C      CLOSE (LUN)
C
C      RETURN
C      END
C      SUBROUTINE SETALM(ALAM,IEMAX,IEDIM,IFLAV,ALMDEL)
C
C      THIS SUBROUTINE READS THE NUMBER OF ENERGY CHANNELS AND THEIR
C      ENERGIES AT GEOSYNCHRONOUS ORBIT AND CALCULATES THE ENERGY
C      INVARIANTS NEEDED TO RUN THE PROGRAM.
C      THE INPUT ENERGY IS IN EV AND MUST BE IN ASCENDING ORDER SO THAT THE
C      ALMDEL VARIABLES CAN BE CORRECTLY CALCULATED.
C
C      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LIUDAT,LUHDYE,
1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
C      PARAMETER (VM6=7.0)
C
C      DIMENSION ALAM(IEDIM)
C      DIMENSION IFLAV(IEDIM)
C      DIMENSION ALMDEL(IEDIM)
C      DIMENSION NUM(3)
C
C      OPEN (UNIT=LUENCH,FILE='ENCHAN',STATUS='OLD')
C
C      READ (LUENCH,*) IEMAX
C      IF (IEMAX.GT.IEDIM) THEN
C          WRITE(6,*) ' NUMBER OF ENERGY CHANNELS GREATER THAN MAXIMUM'
C          STOP
C      ENDIF
C
C      DO 5 N=1,3
C          NUM(N)=0
5      CONTINUE
C
C      DO 10 IE=1,IEMAX
C          READ (LUENCH,*) IFLAV(IE),ENCHNL
C          ISP=IFLAV(IE)
C          NUM(ISP)=NUM(ISP)+1
C          ALAM(IE)= ENCHNL/VM6
C          IF (IFLAV(IE).EQ.1) ALAM(IE)=-ALAM(IE)
C          IF (NUM(ISP).GT.1.AND.(ABS(ALAM(IE)).LT.ABS(ALAM(IE-1)))) THEN
C              WRITE(6,*) ' ENERGY CHANNELS NOT IN INCREASING ORDER'
C              STOP
C          ENDIF
10      CONTINUE
```

MSM45440
MSM45450
MSM45460
MSM45470
MSM45480
MSM45490
MSM45500
MSM45510
MSM45520
MSM45530
MSM45540
MSM45550
MSM45560
MSM45570
MSM45580
MSM45590
MSM45600
MSM45610
MSM45620
MSM45630
MSM45640
MSM45650
MSM45660
MSM45670
MSM45680
MSM45690
MSM45700
MSM45710
MSM45720
MSM45730
MSM45740
MSM45750
MSM45760
MSM45770
MSM45780
MSM45790
MSM45800
MSM45810
MSM45820
MSM45830
MSM45840
MSM45850
MSM45860
MSM45870
MSM45880
MSM45890
MSM45900
MSM45910
MSM45920
MSM45930
MSM45940
MSM45950
MSM45960
MSM45970
MSM45980
MSM45990
MSM46000
MSM46010
MSM46020
MSM46030
MSM46040
MSM46050
MSM46060
MSM46070

90/06/30
12:03:44

msml.for

73

```
C
C COMPUTE ALMDEL
  DO 20 IE=1,IEMAX
    ISP=IFLAV(IE)
    IF(NUM(ISP).GE.2) THEN
      IF(IE.EQ.1.OR.IFLAV(IE-1).NE.ISP) THEN
        ALMMIN=0.
      ELSE
        ALMMIN=(ALAM(IE-1)+ALAM(IE))/2.
      END IF
    ELSE
      IF(IE.EQ.IEMAX.OR.IFLAV(IE+1).NE.ISP) THEN
        ALMMAX=1.5*ALAM(IE)-.5*ALAM(IE-1)
      ELSE
        ALMMAX=(ALAM(IE)+ALAM(IE+1))/2.
      END IF
    END IF
    ALMDEL(IE)=ABS(ALMMAX)-ABS(ALMMIN)
  END IF
20 CONTINUE
C
C
C WRITE(6,997)
997  FORMAT(1H1,'K   ALAM(K)           ALMDEL(K)')
  DO 30 K=1,IEMAX
    WRITE(6,996) K,ALAM(K),ALMDEL(K)
996  FORMAT(1H ,I3,2F15.4)
30   CONTINUE
    CLOSE(LUENCH)
    RETURN
  END
SUBROUTINE BNDSET(LATDIM,LTDIM,IEDIM,ITMDIM,ITMMAX,
1   IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,FLXMAT,
2   NAUGEL,TIMTAG,BNDLOC,IFLAV,AUGPAR,R,P,VM,ETA,ETABND,
3   FLXBND,ALAM,ALMDEL,IEMAX)
C
C
C VERSION: 1.00
C
C DATE: 09.02.89
C       05.08.90
C
C PURPOSE: SUBROUTINE TO COMPUTE BNDY PLASMA DISTRIBUTION BASED ON
C           STATISTICAL (KP-BASED) DATA. THE
C           GENERAL SCHEME IS DESCRIBED IN SECTION 2.5.5 OF
C           CONTRACT #F19c28-87-K-0001 FINAL REPORT.
C
C INPUT:
C   LATDIM      NUMBER OF LATITUDINAL GRID PTS
C   LTDIM       NUMBER OF LOCAL TIME GRID POINTS (INCL WRAP)
C   IEDIM       NUMBER OF ENERGY SPECIES
C   ITMDIM      MAX NUMBER OF TEMPORAL GRID PTS
C   ITMMAX      ACTUAL NUMBER
C               THIS FUN
C   IRDIM       NUMBER OF R BINS IN FLXMAT ARRAY
C   NRGDIM      NUMBFR OF LOGARITHMICALLY SPACE ENERGY BINS
C               IN FLXMAT ARRAY
C   KPDIM       NUMBER OF KP BINS IN FLXMAT ARRAY
C   KPPLUS      AUGMENTED KP DIMENSION OF FLXMAT (KPDIM+2)
C   ISPDIM      NUMBER OF MASS SPECIES IN FLXMAT ARRAY
C   FLXR        LOG(R) VALUES FOR WHICH FLXMAT IS COMPUTED (RE)
C   FLXNRG      LOG OF ENERGY VALUES FOR WHICH FLXMAT IS
C               COMPUTED (LOG10(EV))
C   FLXKP       KP VALUES FOR WHICH FLXMAT IS COMPUTED
```

MSM46080
MSM46090
MSM46100
MSM46110
MSM46120
MSM46130
MSM46140
MSM46150
MSM46160
MSM46170
MSM46180
MSM46190
MSM46200
MSM46210
MSM46220
MSM46230
MSM46240
MSM46250
MSM46260
MSM46270
MSM46280
MSM46290
MSM46300
MSM46310
MSM46320
MSM46330
MSM46340
MSM46350
MSM46360
MSM46370
MSM46380
MSM46390
MSM46400
MSM46410
MSM46420
MSM46430
MSM46440
MSM46450
MSM46460
MSM46470
MSM46480
MSM46490
MSM46500
MSM46510
MSM46520
MSM46530
MSM46540
MSM46550
MSM46560
MSM46570
MSM46580
MSM46590
MSM46600
MSM46610
MSM46620
MSM46630
MSM46640
MSM46650
MSM46660
MSM46670
MSM46680
MSM46690
MSM46700
MSM46710

90/06/30
12:03:44

msml.for

74

```
C      FLXMAT      ARRAY OF KP-DEPENDENT LOG(FLUX) VALUES      MSM46720
C      AS FCN OF R AND ENERGY (LOG10(PARTICLES/CM**2/      MSM46730
C      S/SR/EV))      MSM46740
C      NAUGEL      DIMENSION OF AUGMENTED PARRAY, AUGPAR      MSM46750
C      TIMTAG      STRING GIVING TIMES FOR WHICH E AND B FIELD      MSM46760
C      VALUES ARE CALCULATED      MSM46770
C      BNDLOC      ARRAY GIVING BNDY LOCATIONS      MSM46780
C      IFLAV      STRING THAT GIVES WHAT KIND OF PARTICLE      MSM46790
C      IFLAV=1 FOR ELECTRONS, IFLAV=2 FOR H+,      MSM46800
C      IFLAV=3 FOR O+ IONS      MSM46810
C      AUGPAR      AUGMENTED PARAMETER ARRAY      MSM46820
C      R      RADIAL DISTANCE TO BMIN PT (RE)      MSM46830
C      P      HOUR ANGLE MEASURED EASTWARD FROM NOON IN X-Y      MSM46840
C      PLANE OF MAGNETOSPHERE (RADIAN)      MSM46850
C      VM      (FLUX TUBE VOLUME)**-2/3 (RE/NT)**-2/3      MSM46860
C      ETA      INVARIANT DENSITY ARRAY      MSM46870
C      ALAM      STRING GIVING ENERGY INVARIANTS      MSM46880
C      ALMDEL      STRING GIVING ENERGY INVARIANT INTERVALS      MSM46890
C      IEMAX      NUMBER OF ENERGY SPECIES FOR RUN      MSM46900
C      MSM46910
C      MSM46920
C      MSM46930
C      OUTPUT:      MSM46940
C      ETABND      BOUNDARY ETA DISTRIBUTION      MSM46950
C      FLXBND      BOUNDARY FLUX VALUES      MSM46960
C      MSM46970
C      MSM46980
C      MSM46990
C      PROGRAMMER: R.W. SPIRO      MSM47000
C      MSM47010
C      MSM47020
C      MSM47030
C      MSM47040
C      COMMON /IRDREC/ IRECPT,IRECEB,IRDBEG      MSM47050
C      MSM47060
C      INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,      MSM47070
C      1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFYZ,IMFBZ,CLAPSE,      MSM47080
C      2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,      MSM47090
C      3      STAND,PCP,DEQDT,IPATT      MSM47100
C      MSM47110
C      COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,      MSM47120
C      1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFYZ,IMFBZ,CLAPSE,      MSM47130
C      2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,      MSM47140
C      3      STAND,PCP,IPATT,DDST,DEQDT      MSM47150
C      MSM47160
C      MSM47170
C      DIMENSION VM(LATDIM,LTDIM,ITMDIM),ETA(LATDIM,LTDIM,IEDIM)      MSM47180
C      DIMENSION R(LATDIM,LTDIM,ITMDIM),P(LATDIM,LTDIM,ITMDIM)      MSM47190
C      DIMENSION BNDLOC(LTDIM,ITMDIM),TIMTAG(ITMDIM),IFLAV(IEDIM)      MSM47200
C      DIMENSION ETABND(LTDIM,IEDIM),ALAM(IEDIM),ALMDEL(IEDIM)      MSM47210
C      DIMENSION FLXBND(LTDIM,IEDIM,ITMDIM)      MSM47220
C      DIMENSION AUGPAR(NAUGEL,ITMDIM)      MSM47230
C      MSM47240
C      DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)      MSM47250
C      DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)      MSM47260
C      MSM47270
C      DATA VM13 /2.182/      MSM47280
C      MSM47290
C      PI=ATAN2(0.,-1.)      MSM47300
C      WRITE(6,*) 'IN BNDSET'      MSM47310
C      MSM47320
C      MAIN DO LOOPS      MSM47330
C      ITBEG=1      MSM47340
C      MSM47350
```

90/06/30
12:03:44

msm1.for

75

```
      ITEND=ITMMAX
C
      VMBND=VM13
      RBND=13.
      PBND=PI
C
C   LOOP OVER TIME LABELS
      DO 5 IT=ITBEG,ITEND
C
C   LOOP OVER LOCAL TIME COORDINATE
      DO 10 J=2,LTDIM-2
C
      BI=BNDLOC(J,IT)
      BJ=J
      BT=IT
C
C   LOOP OVER ENERGY SPECIES
      DO 20 K=1,IEMAX
      ISP=IFLAV(K)
      IF(IFLAV(K).EQ.2) ATOMWT=1.
      IF(IFLAV(K).EQ.3) ATOMWT=16.
C
C
C
C
      FKP=AUGPAR(KP,IT)
      ENRG=VMBND*ABS(ALAM(K))
C   DEBUG 590
C      IF(J.EQ.2) THEN
C        WRITE(6,*)'IT,ISP,FKP,RBND,PBND,ENRG',IT,ISP,FKP,RBND,
C          2          PBND,ENRG
C      END IF
C
      FLUX=FLXVAL(ISP,FKP,RBND,PBND,ENRG,IRDIM,
          2          NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,
          3          FLXNRG,FLXKP)
C
C
C
C
      FLXBND(J,K,IT)=FLUX
C
C   CONVERT FLUX TO INVARIANT DENSITY
C
      IF(ALAM(K).LT.0.) THEN
          2      ETABND(J,K,IT)=FLUX*ALMDEL(K)/7.392E-16/
              SQRT(ABS(ALAM(K)))/VMBND
      ELSE
          2      ETABND(J,K,IT)=FLUX*ALMDEL(K)*SQRT(ATOMWT)/1.731E-17/
              SQRT(ABS(ALAM(K)))/VMBND
      END IF
C
C
C   20   CONTINUE
C
C   10   CONTINUE
C
      DO 15 K=1,IEMAX
      ETABND(1,K,IT)=ETABND(LTDIM-3,K,IT)
      ETABND(LTDIM-1,K,IT)=ETABND(2,K,IT)
      ETABND(LTDIM,K,IT)=ETABND(3,K,IT)
      FLXBND(1,K,IT)=FLXBND(LTDIM-3,K,IT)
      FLXBND(LTDIM-1,K,IT)=FLXBND(2,K,IT)
```

MSM47360
MSM47370
MSM47380
MSM47390
MSM47400
MSM47410
MSM47420
MSM47430
MSM47440
MSM47450
MSM47460
MSM47470
MSM47480
MSM47490
MSM47500
MSM47510
MSM47520
MSM47530
MSM47540
MSM47550
MSM47560
MSM47570
MSM47580
MSM47590
MSM47600
MSM47610
MSM47620
MSM47630
MSM47640
MSM47650
MSM47660
MSM47670
MSM47680
MSM47690
MSM47700
MSM47710
MSM47720
MSM47730
MSM47740
MSM47750
MSM47760
MSM47770
MSM47780
MSM47790
MSM47800
MSM47810
MSM47820
MSM47830
MSM47840
MSM47850
MSM47860
MSM47870
MSM47880
MSM47890
MSM47900
MSM47910
MSM47920
MSM47930
MSM47940
MSM47950
MSM47960
MSM47970
MSM47980
MSM47990

90/06/30
12:03:44

msml.for

76

FLXBND (LTDIM, K, IT) = FLXBND (3, K, IT)

15 CONTINUE

5 CONTINUE

RETURN

END

FUNCTION CFACT (IDIM, JDIM, IEDIM, IUSE, JUSE, IEUSE, ETAEQK, ALAM,
2 PWR, ETA, ALAME)

FUNCTION SUBPROGRAM TO CALCULAT CORRECTION FACTOR BASED ON
EQUILIBRIUM ETA VALUE.

CFACT = (1. - ETAEQK / ETA)

RETURN

END

SUBROUTINE SETSCT (L, ISEC, IEDIM, LTDIM, ITMDIM, LATDIM, FKP, ALAM,
2 ALOCT, SCTFRC, IEMAX)

VERSION 2.00

DATE: 09.11.89

PURPOSE:

SUBROUTINE TO COMPUTE SPECIES-DEPENDENT PITCH
ANGLE SCATTERING EFFICIENCY SCTFRC (IE, J, L).

PROGRAMMER: R.W. SPIRO

REF: SECTION 2.6.3 OF THE FINAL REPORT FOR
CONTRACT #F19628-87-K-0001

DIMENSION SCTFRC (IEDIM, LTDIM, ITMDIM)
DIMENSION ALOCT (LATDIM, LTDIM), ALAM (IEDIM)

PI = ATAN2 (0., -1.)

LOOP OVER ALL LOCAL TIME GRID LINES
DO 10 J = 1, LTDIM

LOOP OVER ALL ENERGY SPECIES

DO 20 IE = 1, IEMAX

IF (ALAM (IE) .LT. 3000.) THEN

USE KP DEPENDENT FORM, INDEPENDENT OF LOCAL TIME

IF (FKP .LE. 1.5) THEN

SCTFRC (IE, J, L) = .333

ELSE IF (FKP .GT. 1.5 .AND. FKP .LE. 2.5) THEN

SCTFRC (IE, J, L) = .333 * (2.5 - FKP) + .667 * (FKP - 1.5)

ELSE

SCTFRC (IE, J, L) = .667

END IF

ELSE

USE LOCAL TIME DEPENDENT FORM, INDEPENDENT OF KP

IF (ALOCT (LATDIM, J) .GE. PI / 2. .AND. ALOCT (LATDIM, J) .LE.

3. * PI / 2.) THEN

SCTFRC (IE, J, L) = 0.1 / SQRT (ABS (ALAM (IE)) / 3000.)

ELSE

SCTFRC (IE, J, L) = (0.1 + 0.9 * COS (ALOCT (LATDIM, J))) /

SQRT (ABS (ALAM (IE)) / 3000.)

END IF

END IF

MSM48000

MSM48010

MSM48020

MSM48030

MSM48040

MSM48050

MSM48060

MSM48070

MSM48080

MSM48090

MSM48100

MSM48110

MSM48120

MSM48130

MSM48140

MSM48150

MSM48160

MSM48170

MSM48180

MSM48190

MSM48200

MSM48210

MSM48220

MSM48230

MSM48240

MSM48250

MSM48260

MSM48270

MSM48280

MSM48290

MSM48300

MSM48310

MSM48320

MSM48330

MSM48340

MSM48350

MSM48360

MSM48370

MSM48380

MSM48390

MSM48400

MSM48410

MSM48420

MSM48430

MSM48440

MSM48450

MSM48460

MSM48470

MSM48480

MSM48490

MSM48500

MSM48510

MSM48520

MSM48530

MSM48540

MSM48550

MSM48560

MSM48570

MSM48580

MSM48590

MSM48600

MSM48610

MSM48620

MSM48630

90/06/30
12:03:44

msm1.for

77

```
C
20    CONTINUE
10    CONTINUE
C
C
C    RETURN
C    END
C    SUBROUTINE PFIX(P1,P2,P3,P4,P5)
C
C    PURPOSE:  SUBROUTINE TO ADJUST MODULUS OF P2 AND P3 TO MATCH P1.
C              ON OUTPUT P4 CORRESPONDS TO P2, P5 TO P3
C
C    VERSION 1.00                      DATE: 09.04.89
C
C    PROGRAMMER: R.W. SPIRO
C
C    PI=ATAN2(0.,-1.)
C
C    P4=P2
C    P5=P3
C
C    IF (ABS(P2-P3).GT.PI) THEN
C      IF (P2.LT.P3) THEN
C        P4=P2+2.*PI
C      ELSE
C        P5=P3+2.*PI
C      END IF
C
C    IF (P1.LT.PI) THEN
C      P4=P4-2.*PI
C      P5=P5-2.*PI
C    END IF
C
C    END IF
C
C    RETURN
C    END
C    SUBROUTINE INITAL(LL,LATDIM,LTDIM,IEDIM,ITMDIM,IRDIM,
2      NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXR,FLXNRG,FLXKP,
3      FLXMAT,IFLAV,FKP,R,P,VM,ETA,ETABEG,ALAM,ALMDEL,
4      BNDLOC,AZERO,BZERO,
5      CZERO,IEMAX)
C
C    VERSION: 1.00                      DATE: 10.10.89
C
C    PURPOSE: SUBROUTINE TO COMPUTE INITIAL PLASMA DISTRIBUTION BASED ON
C              EITHER STATISTICAL (KP-BASED) DATA OR RESULTS
C              FROM PREVIOUS RUNS.  THE GENERAL SCHEME IS DESCRIBED IN
C              SECTION 2.5.4 OF THE FINAL REPORT FOR
C              CONTRACT #F19628-87-K-0001
C
C    INPUT:
C      LL          TEMPORAL GRID INDEX
C      LATDIM      NUMBER OF LATITUDINAL GRID PTS
C      LTDIM       NUMBER OF LOCAL TIME GRID POINTS (INCL WRAP)
C      IEDIM       MAX NUMBER OF ENERGY SPECIES
C      ITMDIM      MAX NUMBER OF TEMPORAL GRID PTS
C      IRDIM       R DEPTH IN FLXMAT ARRAY
C      NRGDIM      ENERGY DEPTH IN FLXMAT ARRAY
C      KPDIM       KP DEPTH IN FLXMAT ARRAY
C      KPPLUS      AUGMENTED KP DIMENSION OF FLXMAT (KPDIM+2)
```

MSM48640
MSM48650
MSM48660
MSM48670
MSM48680
MSM48690
MSM48700
MSM48710
MSM48720
MSM48730
MSM48740
MSM48750
MSM48760
MSM48770
MSM48780
MSM48790
MSM48800
MSM48810
MSM48820
MSM48830
MSM48840
MSM48850
MSM48860
MSM48870
MSM48880
MSM48890
MSM48900
MSM48910
MSM48920
MSM48930
MSM48940
MSM48950
MSM48960
MSM48970
MSM48980
MSM48990
MSM49000
MSM49010
MSM49020
MSM49030
MSM49040
MSM49050
MSM49060
MSM49070
MSM49080
MSM49090
MSM49100
MSM49110
MSM49120
MSM49130
MSM49140
MSM49150
MSM49160
MSM49170
MSM49180
MSM49190
MSM49200
MSM49210
MSM49220
MSM49230
MSM49240
MSM49250
MSM49260
MSM49270

90/06/30
12:03:44

78

msm1.for

```
C      ISPDIM      NUMBER OF MASS SPECIES      MSM49280
C      FLXR        ALOG10(R) VALUES FOR WHICH FLXMAT ARRAY IS      MSM49290
C                  CALCULATED      MSM49300
C      FLXNRG      ALOG10(ENERGY) VALUES FOR WHICH FLXMAT ARRAY IS      MSM49310
C                  CALCULATED      MSM49320
C      FLXKP       KP VALUES FOR WHICH FLXMAT ARRAY IS CALCULATED      MSM49330
C      FLXMAT      ALOG10(FLUX) FOR A GIVEN R,ENERGY, AND KP      MSM49340
C      IFLAV       STRING THAT GIVES WHAT KIND OF PARTICLE      MSM49350
C                  IFLAV=1 FOR ELECTRONS, IFLAV=2 FOR H+,      MSM49360
C                  IFLAV=3 FOR O+ IONS      MSM49370
C      FKP         FLOATING PT KP VALUE      MSM49380
C      R           RADIAL DISTANCE TO BMIN PT (RE)      MSM49390
C      P           HOUR ANGLE MEASURED EASTWARD FROM NOON IN X-Y      MSM49400
C                  PLANE OF MAGNETOSPHERE (RADIAN)      MSM49410
C      VM          (FLUX TUBE VOLUME)**-2/3 (RE/NT)**-2/3      MSM49420
C      ETA         INVARIANT DENSITY ARRAY      MSM49430
C      ALAM        STRING GIVING ENERGY INVARIANTS      MSM49440
C      ALMDEL      STRING GIVING ENERGY INVARIANT INTERVALS      MSM49450
C      BNDLOC      ARRAY GIVING BNDY LOCATIONS      MSM49460
C      AZERO       STATISTICAL MODEL WEIGHTING PARAMETER      MSM49470
C      BZERO       OLD RESULTS WEIGHTING PARAMETER      MSM49480
C      CZERO       TRACEBACK WEIGHTING PARAMETER      MSM49490
C      IEMAX       NUMBER OF PARTICLE SPECIES USED      MSM49500
C                  MSM49510
C                  MSM49520
C      WEIGHTING PARAMETERS SET IN DATA STATEMENTS      MSM49530
C      GRZERO      PARAMETER TO SPECIFY R GAUSSIAN WIDTH      MSM49540
C      GPZERO      PARAMETER TO SPECIFY P GAUSSIAN WIDTH      MSM49550
C      GLZERO      PARAMETER TO SPECIFY ALAM GAUSSIAN WIDTH      MSM49560
C                  MSM49570
C                  MSM49580
C      OUTPUT:     MSM49590
C      ETABEG      INITIAL ETA DISTRIBUTION      MSM49600
C                  MSM49610
C                  MSM49620
C                  MSM49630
C      PROGRAMMER: R.W. SPIRO      MSM49640
C                  MSM49650
C                  MSM49660
C                  MSM49670
C                  MSM49680
C      COMMON /IRDREC/ IRECPT,IREECB,IRDBEG      MSM49690
C                  MSM49700
C                  MSM49710
C      DIMENSION VM(LATDIM,LTDIM,ITMDIM),ETA(LATDIM,LTDIM,IEDIM)      MSM49720
C      DIMENSION R(LATDIM,LTDIM,ITMDIM),P(LATDIM,LTDIM,ITMDIM)      MSM49730
C      DIMENSION IFLAV(IEDIM)      MSM49740
C      DIMENSION ETABEG(LATDIM,LTDIM,IEDIM),ALAM(IEDIM),ALMDEL(IEDIM)      MSM49750
C      DIMENSION BNDLOC(LTDIM,ITMDIM)      MSM49760
C                  MSM49770
C      DIMENSION FLXMAT(IRDIM,NRGDIM,KPPLUS,ISPDIM)      MSM49780
C      DIMENSION FLXR(IRDIM),FLXNRG(NRGDIM),FLXKP(KPDIM)      MSM49790
C                  MSM49800
C      DATA GRZERO/.05/,GPZERO/.05/,GLZERO/.3/      MSM49810
C                  MSM49820
C                  MSM49830
C      PI=ATAN2(0.,-1.)      MSM49840
C                  MSM49850
C      INITIALIZE ETABEG ARRAY      MSM49860
C      DO 2 K=1,IEMAX      MSM49870
C      DO 4 J=1,LTDIM      MSM49880
C      DO 6 I=1,LATDIM      MSM49890
C      ETABEG(I,J,K)=0.      MSM49900
C      6      CONTINUE      MSM49910
```

90/06/30
12:03:44

msml.for

79

```

      4      CONTINUE
      2      CONTINUE
C
C MAIN DO LOOPS
C
      DO 10 K=1,IEMAX
        ISP=IFLAV(K)
        IF(IFLAV(K).EQ.2) ATOMWT=1.
        IF(IFLAV(K).EQ.3) ATOMWT=16.
        DO 20 J=1,LTDIM
          IBEG=BNDLOC(J,LL)
          DO 30 I=IBEG,LATDIM
            SUM1=0.
            SUM2=0.
            ENRG=ABS( ALAM(K) ) * VM(I,J,LL)
C
C
C
C PREVIOUS RESULT COMPONENT OF SUM
          IF(ETA(I,J,K).GT.0.) THEN
            IF( ALAM(K) .LT.0.) THEN
              FLXOLD=7.392E-16*SQRT(ABS( ALAM(K) ) ) * VM(I,J,LL) *
2              ETA(I,J,K) / ALMDEL(K)
            ELSE
              FLXOLD=1.731E-17*SQRT(ABS( ALAM(K) ) ) * VM(I,J,LL) *
2              ETA(I,J,K) / SQRT(ATOMWT) / ALMDEL(K)
            END IF
C
            SUM1=SUM1+BZERO
            SUM2=SUM2+BZERO*FLXOLD
C
          END IF
C
C STATISTICAL (KP-BASED) MODEL COMPONENT OF SUM
C
          SUM1=SUM1+AZERO
          SUM2=SUM2+AZERO*FLXVAL( ISP,FKP,R(I,J,LL),P(I,J,LL),ENRG,
2          IRDIM,NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,FLXR,
3          FLXNRG,FLXKP)
C
C CALCULATE DIFFERENTIAL FLUX
C
          IF(SUM1.GT.0.) THEN
            FLUX=SUM2/SUM1
          ELSE
            FLUX=0.
          END IF
C
C CONVERT FLUX TO INVARIANT DENSITY
C
          IF( ALAM(K) .LT.0.) THEN
            ETABEG(I,J,K)=FLUX*ALMDEL(K) / 7.392E-16 /
2            SQRT(ABS( ALAM(K) ) ) / VM(I,J,LL)
          ELSE
            ETABEG(I,J,K)=FLUX*ALMDEL(K) * SQRT(ATOMWT) / 1.731E-17 /
2            SQRT(ABS( ALAM(K) ) ) / VM(I,J,LL)
          END IF
C
C
C
30      CONTINUE
C
```

MSM49920
MSM49930
MSM49940
MSM49950
MSM49960
MSM49970
MSM49980
MSM49990
MSM50000
MSM50010
MSM50020
MSM50030
MSM50040
MSM50050
MSM50060
MSM50070
MSM50080
MSM50090
MSM50100
MSM50110
MSM50120
MSM50130
MSM50140
MSM50150
MSM50160
MSM50170
MSM50180
MSM50190
MSM50200
MSM50210
MSM50220
MSM50230
MSM50240
MSM50250
MSM50260
MSM50270
MSM50280
MSM50290
MSM50300
MSM50310
MSM50320
MSM50330
MSM50340
MSM50350
MSM50360
MSM50370
MSM50380
MSM50390
MSM50400
MSM50410
MSM50420
MSM50430
MSM50440
MSM50450
MSM50460
MSM50470
MSM50480
MSM50490
MSM50500
MSM50510
MSM50520
MSM50530
MSM50540
MSM50550

90/06/30
12:03:44

msm1.for

80

```
DO 32 I=1, IBEG-1
  ETABEG(I, J, K) = ETABEG( IBEG, J, K)
32  CONTINUE

C
C
20  CONTINUE
10  CONTINUE

C
C
  RETURN
  END
  SUBROUTINE TIMINC( ITIME, ITMNEW, IINC )

C
C  VERSION 2.00                      DATE: 09.06.89
C
C  PURPOSE:  INCREMENT TIME FOR NEXT E AND B FIELD RECORD
C
C  PROGRAMMER: R.W. SPIRO
C
C
C      INTEGER IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST, DDST,
1      EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2      GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3      STAND, PCP, DEQDT, IPATT

C
C      COMMON /ARINDX/ IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST,
1      EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2      GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3      STAND, PCP, IPATT, DDST, DEQDT

C
C      DIMENSION ITIME(3), ITMNEW(3), IINC(3)

C
C      ITMNEW( IYEAR ) = ITIME( IYEAR )
C      ITMNEW( IDAY ) = ITIME( IDAY )

C
C      ITMNEW( ISECND ) = ITIME( ISECND ) + IINC( ISECND )

C
C      IF ( ITMNEW( ISECND ) .GE. 86400 ) THEN
C          ITMNEW( ISECND ) = MOD( ITMNEW( ISECND ), 86400 )
C          ITMNEW( IDAY ) = ITMNEW( IDAY ) + 1
C
C      -----> CHANGE BEFORE THE YEAR 2100 *****
C
C      IF ( ITMNEW( IDAY ) .GT. 365 .AND. MOD( ITIME( IYEAR ), 4 ) .NE. 0 ) THEN
C          ITMNEW( IDAY ) = MOD( ITMNEW( IDAY ), 365 )
C          ITMNEW( IYEAR ) = ITMNEW( IYEAR ) + 1
C      ELSEIF ( ITMNEW( IDAY ) .GT. 366 .AND. MOD( ITIME( IYEAR ), 4 ) .EQ. 0 ) THEN
C          ITMNEW( IDAY ) = MOD( ITMNEW( IDAY ), 366 )
C          ITMNEW( IYEAR ) = ITMNEW( IYEAR ) + 1
C      ENDIF

C
C      END IF

C
C      RETURN
C      END

C
C
C
C      SUBROUTINE SETREF( LATDIM, LTDIM, IEDIM, ITMDIM, ITMMAX,
2      IRDIM, NRGDIM, KPDIM, KPPLUS, ISPDIM, TIMTAG, TVAL, IFLAV,
```

MSM50560
MSM50570
MSM50580
MSM50590
MSM50600
MSM50610
MSM50620
MSM50630
MSM50640
MSM50650
MSM50660
MSM50670
MSM50680
MSM50690
MSM50700
MSM50710
MSM50720
MSM50730
MSM50740
MSM50750
MSM50760
MSM50770
MSM50780
MSM50790
MSM50800
MSM50810
MSM50820
MSM50830
MSM50840
MSM50850
MSM50860
MSM50870
MSM50880
MSM50890
MSM50900
MSM50910
MSM50920
MSM50930
MSM50940
MSM50950
MSM50960
MSM50970
MSM50980
MSM50990
MSM51000
MSM51010
MSM51020
MSM51030
MSM51040
MSM51050
MSM51060
MSM51070
MSM51080
MSM51090
MSM51100
MSM51110
MSM51120
MSM51130
MSM51140
MSM51150
MSM51160
MSM51170
MSM51180
MSM51190

90/06/30
12:03:44

msm1.for

81

```

3      VM, R, P, BNDLOC, ALAM, ALMDEL, FLXMAT, FLXR, FLXNRG,
4      FLXKP, FLXREF, IEMAX)
C
C  VERSION 2.00                      DATE: 10.10.89
C                                     05.08.90
C
C  PURPOSE:
C    SUBROUTINE TO SET UPPER LIMIT REFERENCE FLUX FOR TIME TVAL
C
C  PROGRAMMER: R.W. SPIRO
C
C  REF:  SECTION 2.5.6 OF THE FINAL REPORT FOR
C        CONTRACT #F19628-87-K-0001
C
C    LATDIM      NUMBER OF LATITUDINAL GRID SPACES
C    LTDIM       NUMBER OF LOCAL TIME (LONGITUDINAL) GRID SPACES
C    IEDIM       MAXIMUM NUMBER OF ENERGY SPECIES
C    ITMDIM      MAXIMUM NUMBER OF TEMPORAL STEPS
C    ITMMAX      ACTUAL NUMBER OF TEMPORAL STEPS FOR THIS RUN
C    IRDIM       R DIMENSION OF FLXMAT ARRAYS
C    NRGDIM      ENERGY DIMENSION OF FLXMAT ARRAYS
C    KPDIM       KP DIMENSION OF FLXMAT ARRAYS
C    KPPLUS      KP DIMENSION OF AUGMENTED FLXMAT ARRAYS
C                (KPPLUS=KPDIM+2)
C    ISPDIM      PLASMA MASS SPECIES DIMENSION
C    TIMTAG      VECTOR GIVING TIME AT WHICH E AND B FIELD PARAMETERS
C                ARE COMPUTED (SECONDS)
C    TVAL        CURRENT TIME (SECONDS)
C    IFLAV       MASS SPECIES FOR A GIVEN PARTICLE ENERGY SPECIES
C                1=ELECTRONS, 2=H+ IONS, 3=O+ IONS
C    VM          (FLUX TUBE VOLUME)**(-2/3)
C    R           ARRAY GIVING RADIAL DISTANCE OF GRID POINTS IN
C                MAGNETOSPHERIC EQUATORIAL PLANE (RE)
C    P           LOCAL TIME HOUR ANGLE MEASURED EASTWARD FROM NOON
C                OF GRID PTS IN MAGNETOSPHERIC EQUATORIAL PLANE
C                (RADIAN)
C    BNDLOC      LOCATION OF OUTER BOUNDARY OF DETAILED MODEL
C    ALAM        ENERGY INVARIANT VECTOR (EV (RE/NT)**2/3)
C    ALMDEL      WIDTH OF ENERGY INVARIANT CHANNELS
C    FLXMAT      ARRAY OF EMPIRICAL FLUXES AS FUNCTION OF R, ENERGY,
C                AND KP
C    FLXR        R VALUES AT WHICH FLXMAT IS CALCULATED
C    FLXNRG      ENERGY VALUES AT WHICH FLXMAT IS CALCULATED
C    FLXKP       KP VALUES AT WHICH FLXMAT IS CALCULATED
C    IEMAX       ACTUAL NUMBER OF ENERGY CHANNELS USED
C
C
C  OUTPUT:
C    FLXREF      UPPER LIMIT REFERENCE FLUX (LOG(#/CM**2 S EV))
C
C    DIMENSION TIMTAG(ITMDIM), ALAM(IEDIM), ALMDEL(IEDIM)
C    DIMENSION VM(LATDIM, LTDIM, ITMDIM), R(LATDIM, LTDIM, ITMDIM)
C    DIMENSION P(LATDIM, LTDIM, ITMDIM), FLXREF(LATDIM, LTDIM, IEDIM)
C    DIMENSION IFLAV(IEDIM)
C    DIMENSION BNDLOC(LTDIM, ITMDIM)
C    DIMENSION FLXMAT(IRDIM, NRGDIM, KPPLUS, ISPDIM)
C    DIMENSION FLXR(IRDIM), FLXNRG(NRGDIM), FLXKP(KPDIM)
C
C
C  CALCULATE NORMALIZED TIME BT
C    BT=TNORML(TVAL, TIMTAG, ITMDIM)
C    ITT=BT
C    FT=REAL(ITT)
C
```

MSM51200
MSM51210
MSM51220
MSM51230
MSM51240
MSM51250
MSM51260
MSM51270
MSM51280
MSM51290
MSM51300
MSM51310
MSM51320
MSM51330
MSM51340
MSM51350
MSM51360
MSM51370
MSM51380
MSM51390
MSM51400
MSM51410
MSM51420
MSM51430
MSM51440
MSM51450
MSM51460
MSM51470
MSM51480
MSM51490
MSM51500
MSM51510
MSM51520
MSM51530
MSM51540
MSM51550
MSM51560
MSM51570
MSM51580
MSM51590
MSM51600
MSM51610
MSM51620
MSM51630
MSM51640
MSM51650
MSM51660
MSM51670
MSM51680
MSM51690
MSM51700
MSM51710
MSM51720
MSM51730
MSM51740
MSM51750
MSM51760
MSM51770
MSM51780
MSM51790
MSM51800
MSM51810
MSM51820
MSM51830

MSM51840
MSM51850
MSM51860
MSM51870
MSM51880
MSM51890
MSM51900
MSM51910
MSM51920
MSM51930
MSM51940
MSM51950
MSM51960
MSM51970
MSM51980
MSM51990
MSM52000
MSM52010
MSM52020
MSM52030
MSM52040
MSM52050
MSM52060
MSM52070
MSM52080
MSM52090
MSM52100
MSM52110
MSM52120
MSM52130
MSM52140
MSM52150
MSM52160
MSM52170
MSM52180
MSM52190
MSM52200
MSM52210
MSM52220
MSM52230
MSM52240
MSM52250
MSM52260
MSM52270
MSM52280
MSM52290
MSM52300
MSM52310
MSM52320
MSM52330
MSM52340
MSM52350
MSM52360
MSM52370
MSM52380
MSM52390
MSM52400
MSM52410
MSM52420
MSM52430
MSM52440
MSM52450
MSM52460
MSM52470

90/06/30
12:03:44

msml.for

83

```
C
C
C
C
C
C      DIMENSION R(LATDIM,LTDIM,ITMDIM),ALOCT(LATDIM,LTDIM)
C
C
C      PI=ATAN2(0.,-1.)
C
C
C
C
C
C
C
C
C      SEARCH GRID AND TRY TO LOCALIZE (RVAL,XLT) WITHIN A GRID SQUARE
C      DO 30 J=2,LTDIM-2
C        JJ=J
C
C        DO 40 I=2,LATDIM
C          II=I
C
C          GET XLT,ALOCT(I,J), AND ALOCT(I,J+1) IN SAME MODULUS
C          CALL PFIX(XLT,ALOCT(I,J),ALOCT(I,J+1),P4,P5)
C          WRITE(6,*) I,J,RVAL,R(I-1,J,LL),R(I,J,LL),XLT,P4,P5
C
C          IF(RVAL.LE.R(I-1,J,LL)
C 2            .AND.
C 3            RVAL.GT.R(I,J,LL)
C 4            .AND.
C 5            XLT.GE.P4
C 6            .AND.
C 7            XLT.LT.P5)
C 8            THEN
C              GO TO 50
C            END IF
C
C 40  CONTINUE
C 30  CONTINUE
C
C      WRITE(6,*) 'STOPPING IN EFLOC'
C      WRITE(6,*) 'UNABLE TO FIND (BI,BJ)'
C      WRITE(6,*) 'RVAL= ',RVAL,' XLT= ',XLT
C      STOP
C
C
C
C 50  CONTINUE
C
C      DO 45 IK=2,LATDIM
C        IIK=IK
C        IF(RVAL.LE.R(IK-1,J+1,LL).AND.RVAL.GT.R(IK,J+1,LL)) GO TO 60
C 45  CONTINUE
C
C      WRITE(6,*) 'STOPPING IN EFLOC'
C      WRITE(6,*) 'UNABLE TO FIND (BI,BJ)'
C      WRITE(6,*) 'RVAL= ',RVAL,' XLT= ',XLT
C      STOP
C
C
C 60  CONTINUE
C
C
C
C      COMPUTE COEFFICIENTS FOR INTERPOLATION
C      F1=(RVAL-R(II-1,JJ,LL))/(R(II,JJ,LL)-R(II-1,JJ,LL))+FLOAT(II-1)
```

MSM52480
MSM52490
MSM52500
MSM52510
MSM52520
MSM52530
MSM52540
MSM52550
MSM52560
MSM52570
MSM52580
MSM52590
MSM52600
MSM52610
MSM52620
MSM52630
MSM52640
MSM52650
MSM52660
MSM52670
MSM52680
MSM52690
MSM52700
MSM52710
MSM52720
MSM52730
MSM52740
MSM52750
MSM52760
MSM52770
MSM52780
MSM52790
MSM52800
MSM52810
MSM52820
MSM52830
MSM52840
MSM52850
MSM52860
MSM52870
MSM52880
MSM52890
MSM52900
MSM52910
MSM52920
MSM52930
MSM52940
MSM52950
MSM52960
MSM52970
MSM52980
MSM52990
MSM53000
MSM53010
MSM53020
MSM53030
MSM53040
MSM53050
MSM53060
MSM53070
MSM53080
MSM53090
MSM53100
MSM53110

90/06/30
12:03:44

msml.for

84

```

      F2=(RVAL-R(IIK-1,JJ+1,LL))/(R(IIK,JJ+1,LL)-R(IIK-1,JJ+1,LL))+
      2  FLOAT(IIK-1)
C
      CALL PFIX(XLT,ALOCT(1I,JJ),ALOCT(IIK,JJ+1),P4,P5)
C
      BJ=(XLT-P4)/(P5-P4)+FLOAT(JJ)
C
      BI=F1+(BJ-FLOAT(JJ))*(F2-F1)/(FLOAT(JJ+1)-FLOAT(JJ))
C
C
      RETURN
      END
C
C
C
      SUBROUTINE EFBNDY(LATDIM,LTDIM,ITMDIM,LL,EQEDG,STDOFF,VDROP,R,
      2  ALOCT,COLAT,A,B,DX,DY)
C
C  VERSION 1.00                      DATE: 09.13.89
C      1.01                      10.25.89
C  PURPOSE:
C  SUBROUTINE TO CALCULATE LATITUDES AND WIDTHS OF EFIELD BNDYS 2 & 3
C
C  PROGRAMMER: R.W. SPIRO
C
C  REF: SECTION 2.4.2 OF THE FINAL REPORT FOR
C      CONTRACT #F19628-87-K-0001
C
C
      DIMENSION R(LATDIM,LTDIM,ITMDIM),ALOCT(LATDIM,LTDIM)
      DIMENSION COLAT(LATDIM,LTDIM)
      DIMENSION A(3),B(3),DX(3),DY(3)
      DIMENSION XLAMDA(4,3),DLAMDA(4)
C
C
      PI=ATAN2(0.,-1.)
C
      WRITE(6,*) 'IN EFBNDY EQEDG= ',EQEDG,'STDOFF= ',STDOFF
C
      RNOON=.95*STDOFF
      RDAWN=1.4*STDOFF
      RDUSK=1.4*STDOFF
      RMIDNT=2.0*STDOFF
C
C
C  XLAMDA(1,N) REFERS TO NOON; XLAMDA(2,N) REFERS TO DUSK
C  XLAMDA(3,N) REFERS TO MIDNIGHT; XLAMDA(4,N) REFERS TO DAWN
C  WHERE N=1,2,3 REFERS TO THE THREE BNDYS OF THE E FIELD CALCULATION
C  BNDY 1 IS CALCULATED INTERNALLY IN EFIELD
C  BNDYS 2 AND 3 ARE CALCULATED HERE
C
      XLAMDA(3,3)=EQEDG
      XLAMDA(2,3)=EQEDG+1.
      XLAMDA(4,3)=EQEDG+1.
      XLAMDA(1,3)=EQEDG+((66.95-EQEDG)*7.725+2.8*(EQEDG-56.8))/10.15
C
C
C  CALCULATION OF PARAMETERS USED IN CALCULATING BNDY 2
C
C  LOCATE RNOON,RDAWN, RDUSK, AND RMIDNT IN OUR GRID, IE., FIND
C      (BINOON,BJNOON), (BIDUSK,BJDUSK), (BIMDNT,BJMDNT), (BIDAWN,BJDAWN)
C
      TNOON=0.
      CALL EFLOC(RNOON,TNOON,LL,LATDIM,LTDIM,ITMDIM,R,ALOCT,
```

MSM53120
MSM53130
MSM53140
MSM53150
MSM53160
MSM53170
MSM53180
MSM53190
MSM53200
MSM53210
MSM53220
MSM53230
MSM53240
MSM53250
MSM53260
MSM53270
MSM53280
MSM53290
MSM53300
MSM53310
MSM53320
MSM53330
MSM53340
MSM53350
MSM53360
MSM53370
MSM53380
MSM53390
MSM53400
MSM53410
MSM53420
MSM53430
MSM53440
MSM53450
MSM53460
MSM53470
MSM53480
MSM53490
MSM53500
MSM53510
MSM53520
MSM53530
MSM53540
MSM53550
MSM53560
MSM53570
MSM53580
MSM53590
MSM53600
MSM53610
MSM53620
MSM53630
MSM53640
MSM53650
MSM53660
MSM53670
MSM53680
MSM53690
MSM53700
MSM53710
MSM53720
MSM53730
MSM53740
MSM53750

90/06/30
12:03:44

msml.for

85

```

      2          BINOON, BJNOON)
C
      TDUSK=PI/2.
      CALL EFLOC (RDUSK, TDUSK, LL, LATDIM, LTDIM, ITMDIM, R, ALOCT,
      2          BIDUSK, BJDUSK)
C
C
      TMIDNT=PI
      CALL EFLOC (RMIDNT, TMIDNT, LL, LATDIM, LTDIM, ITMDIM, R, ALOCT,
      2          BIMDNT, BJMDNT)
C
      TDAWN=3.*PI/2.
      CALL EFLOC (RDAWN, TDAWN, LL, LATDIM, LTDIM, ITMDIM, R, ALOCT,
      2          BIDAWN, BJDAWN)
C
C
      XLAT1=G3NTRP (COLAT, LATDIM, LTDIM, 1, BINOON, BJNOON, 1.)
      XLAMDA (1, 2) = (PI/2. - XLAT1) * 180. / PI
C
      XLAT2=G3NTRP (COLAT, LATDIM, LTDIM, 1, BIDUSK, BJDUSK, 1.)
      XLAMDA (2, 2) = (PI/2. - XLAT2) * 180. / PI
C
      XLAT3=G3NTRP (COLAT, LATDIM, LTDIM, 1, BIMDNT, BJMDNT, 1.)
      XLAMDA (3, 2) = (PI/2. - XLAT3) * 180. / PI
C
      XLAT4=G3NTRP (COLAT, LATDIM, LTDIM, 1, BIDAWN, BJDAWN, 1.)
      XLAMDA (4, 2) = (PI/2. - XLAT4) * 180. / PI
C
C CHECK BOUNDARY 3 (EQ. EDGE OF SUNWARD FLOW) FOR RIDICULOUS VALUE;
C CHANGE IF NECESSARY
C
      EMAX=.1
      DLAMDA (1)=3.
      DLAMDA (3)=3.
      DLAMDA (2)=4.5E-6*VDROP/EMAX
      DLAMDA (4)=DLAMDA (2)
C
C LOOP OVER NOON, DUSK, MIDNIGHT, AND DAWN
C DO 8 II=1,4
      DELTA=XLAMDA (II, 2) - DLAMDA (II) - XLAMDA (II, 3)
      IF (DELTA.LT.0.) THEN
          WRITE (6, *) 'ADJUSTING XLAMDA (II, 3) IN EFBNDY'
          WRITE (6, *) 'LL=', LL, ' II=', II, ' XLAMDA (OLD) =', XLAMDA (II, 3)
          XLAMDA (II, 3) = XLAMDA (II, 3) + DELTA
          WRITE (6, *) '          XLAMDA (NEW) =', XLAMDA (II, 3)
      END IF
      8 CONTINUE
C
C
      DO 10 II=2,3
          DX (II) = -.5 * (XLAMDA (1, II) - XLAMDA (3, II))
          DY (II) = -.5 * (XLAMDA (2, II) - XLAMDA (4, II))
C
          X = .5 * (180. - XLAMDA (1, II) - XLAMDA (3, II))
          Y = .5 * (180. - XLAMDA (4, II) - XLAMDA (2, II))
C
          A (II) = SQRT ((X*Y)**2 - (DX (II)*DY (II))**2) / (Y**2 - DY (II)**2)
          B (II) = SQRT ((X*Y)**2 - (DX (II)*DY (II))**2) / (X**2 - DX (II)**2)
C
      10 CONTINUE
C
      RETURN
      END
```

MSM53760
MSM53770
MSM53780
MSM53790
MSM53800
MSM53810
MSM53820
MSM53830
MSM53840
MSM53850
MSM53860
MSM53870
MSM53880
MSM53890
MSM53900
MSM53910
MSM53920
MSM53930
MSM53940
MSM53950
MSM53960
MSM53970
MSM53980
MSM53990
MSM54000
MSM54010
MSM54020
MSM54030
MSM54040
MSM54050
MSM54060
MSM54070
MSM54080
MSM54090
MSM54100
MSM54110
MSM54120
MSM54130
MSM54140
MSM54150
MSM54160
MSM54170
MSM54180
MSM54190
MSM54200
MSM54210
MSM54220
MSM54230
MSM54240
MSM54250
MSM54260
MSM54270
MSM54280
MSM54290
MSM54300
MSM54310
MSM54320
MSM54330
MSM54340
MSM54350
MSM54360
MSM54370
MSM54380
MSM54390

90/06/30
12:03:44

msml.for

86

```
C
C ***** F E C O N *****
C
C      SUBROUTINE FECON(ISTART,IINC,IEND,ITMDIM,NAUGEL,MODE,AUGPAR,
1      ITMMAX,TIMTAG)
C
C      VERSION 1.0      DATE: JANUARY 9, 1988
C      2.0      SEPTEMBER 14, 1989
C
C      PROGRAMMER: BRYAN A. BALES
C
C      PURPOSE: FECON CALLS PARGEN TO OBTAIN TIME-NORMALIZED OBSERVED DATA
C      THAT IS THEN USED TO CALCULATE OTHER VALUES NEEDED BY THE
C      EFIELD AND BFIELD ROUTINES.
C
C      PARAMETER (NELTS=28,ITDIM=50)
C
C      INTEGER IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,DDST,
1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
3      STAND,PCP,DEQDT,IPATT
C
C      LOGICAL*1 MODE(NAUGEL,ITMDIM)
C      COMMON /ARINDX/IYEAR,IDAY,ISECND,IPHOUR,IPMIN,IPSEC,KP,DST,
1      EQEDGE,DLATAZ,DLATRV,MLTRV,IMFBX,IMFBY,IMFBZ,CLAPSE,
2      GEOMGX,GEOMGY,GEOMGZ,GEOMGT,LBOUND,SWVEL,SWDEN,TILTW,
3      STAND,PCP,IPATT,DDST,DEQDT
C
C      DIMENSION PARRAY(NELTS,ITDIM),AUGPAR(NAUGEL,ITMDIM),ITIME(3)
C      DIMENSION ITMNEW(3),ISTART(3),IINC(3),IEND(3)
C      DIMENSION TIMTAG(ITMDIM)
C
C      IF (ITDIM.NE.ITMDIM) STOP 'STOPPING IN FECON. ITDIM.NE.ITMDIM'
C
C      DO 30 IDO=1,NAUGEL
C      DO 40 JDO=1,ITMDIM
C      MODE(IDO,JDO)=.FALSE.
40      CONTINUE
30      CONTINUE
C
C      CALL PARGEN(ISTART,IINC,IEND,ITMDIM,NELTS,MODE,PARRAY,
1      ITMMAX,TIMTAG,NAUGEL)
C
C      ITIME(1) = ISTART(1)
C      ITIME(2) = ISTART(2)
C      ITIME(3) = ISTART(3)
C
C      DO 20, ITM=1, ITMMAX
C      DO 10, I=1, NELTS
C      AUGPAR(I,ITM) = PARRAY(I,ITM)
10      CONTINUE
C
C      CALL TILT(ITIME(IDAY),ITIME(IYEAR),AUGPAR(TILTW,ITM))
C
C      CALL STNDL(MODE(STAND,ITM),MODE(SWVEL,ITM),
1      MODE(SWDEN,ITM),PARRAY(SWVEL,ITM),PARRAY(SWDEN,ITM),
2      PARRAY(KP,ITM),AUGPAR(STAND,ITM))
C
C      CALL DSTDFL(MODE(DST,ITM),AUGPAR,NAUGEL,ITMDIM,ITM,ITMMAX,
1      TIMTAG)
C
C      CALL EQTDFL(MODE(EQEDGE,ITM),AUGPAR,NAUGEL,ITMDIM,ITM,
```


90/06/30
12:03:44

msml.for

88

REND = TCONV3 (IEND, ISTART)
NODATA = 0

DO 10, I=6, NELTS

ITM = 1
ITIME (1) = ISTART (1)
ITIME (2) = ISTART (2)
ITIME (3) = ISTART (3)

IF (I.EQ.KP.OR.I.EQ.DST.OR.I.EQ.PCP.OR.I.EQ.IPATT.OR.
I.EQ.CLAPSE.OR.I.EQ.SWDEN.OR.I.EQ.SWVEL.OR.
I.EQ.EQEDGE) THEN

CALL INDATA (PARAM (I), ISTART, IEND, NDIM, DARRY, NUMNUM)
WRITE (6, *) 'COMING OUT OF INDATA, I= ', I, ' NUMNUM= ', NUMNUM
IF ((NUMNUM.EQ.0).AND.(I.EQ.KP)) THEN
WRITE (6, *) ' NO KP AVAILABLE TO RUN PROGRAM.'
STOP
ENDIF

STARTY = REAL (ISTART (1))
IF (ISTART (3).LT.86400) THEN
STARTD = REAL (ISTART (2)) + REAL (ISTART (3)) / 86400.
ELSE IF (ISTART (3).EQ.86400) THEN
STARTD = REAL (ISTART (2))
ELSE
WRITE (6, *) 'INCORRECT START TIME HAS BEEN INPUT'
WRITE (6, *) 'STOPPING PROGRAM IN PARGEN'
STOP
END IF

CALL SMOOTH (DARRY, NDIM, NUMNUM, STARTY, STARTD, NA, XA, YA, IERR)

WRITE (6, *) ' NA= ', NA, ' NUMNUM= ', NUMNUM, ' NDIM= ', NDIM
DO 42 NX=1, NA
WRITE (6, *) ' TIME= ', XA (NX), ' VALUE= ', YA (NX)
CONTINUE

WRITE (6, *) 'RETURNED FROM SMOOTH FOR PARAMETER ', PARAM (I)

IF (IERR.NE.0) THEN
WRITE (6, *) 'TIME REVERSAL IN PARAMETER ', PARAM (I)
STOP 'SERIOUS ERROR IN ENVIRONMENTAL DATA.'
ENDIF

IF (NUMNUM.EQ.0) GO TO 10

CONTINUE

IF (I.EQ.KP) THEN
PARRAY (IYEAR, ITM) = ITIME (IYEAR)
PARRAY (IDAY, ITM) = ITIME (IDAY)
PARRAY (IPHOUR, ITM) = REAL (ITIME (ISECND) / 3600)
PARRAY (IPMIN, ITM) = REAL (MOD (ITIME (ISECND), 3600) / 60)
PARRAY (IPSEC, ITM) = REAL (MOD (ITIME (ISECND), 60))
TIMTAG (ITM) = TCONV3 (ITIME, ISTART)
WRITE (6, *) 'TIMTAG (', ITM, ') = ', TIMTAG (ITM)
ENDIF

IF (I.EQ.IPATT) THEN
CALL DTXIPT (TIMTAG (ITM), NA, XA, YA, Y, DYDX, DELTA, IERR)
PARRAY (I, ITM) = Y

ELSEIF (I.NE.DDST.AND.I.NE.DEQDT) THEN
CALL DTNTRP (TIMTAG (ITM), NA, XA, YA, Y, DYDX, DELTA, IERR)

PARRAY (I, ITM) = Y

MSM55680
MSM55690
MSM55700
MSM55710
MSM55720
MSM55730
MSM55740
MSM55750
MSM55760
MSM55770
MSM55780
MSM55790
MSM55800
MSM55810
MSM55820
MSM55830
MSM55840
MSM55850
MSM55860
MSM55870
MSM55880
MSM55890
MSM55900
MSM55910
MSM55920
MSM55930
MSM55940
MSM55950
MSM55960
MSM55970
MSM55980
MSM55990
MSM56000
MSM56010
MSM56020
MSM56030
MSM56040
MSM56050
MSM56060
MSM56070
MSM56080
MSM56090
MSM56100
MSM56110
MSM56120
MSM56130
MSM56140
MSM56150
MSM56160
MSM56170
MSM56180
MSM56190
MSM56200
MSM56210
MSM56220
MSM56230
MSM56240
MSM56250
MSM56260
MSM56270
MSM56280
MSM56290
MSM56300
MSM56310

90/06/30
12:03:44

msm1.for

89

```

      IF ((DELTA.LE.IGAP(I)).OR.(IGAP(I).EQ.0)) THEN
        MODE(I,ITM) = .TRUE.
      ELSE
        MODE(I,ITM) = .FALSE.
      ENDIF

C
C  DYDX IS IN UNITS OF Y/SEC
C  NEED TO CHANGE TO UNITS OF Y/HOUR
      IF (I.EQ.DST) PARRAY(DDST,ITM) = DYDX*3600.
      IF (I.EQ.EQEDGE) PARRAY(DEQDT,ITM) = DYDX*3600.
      IF (I.EQ.DST)
        1      WRITE(6,*) ' ITM,DST,DYDX ',TIMTAG(ITM),
        1      PARRAY(DST,ITM),PARRAY(DDST,ITM)
      IF (I.EQ.EQEDGE)
        2      WRITE(6,*) ' ITM,EQED,DYDX ',TIMTAG(ITM),
        1      PARRAY(EQEDGE,ITM),PARRAY(DEQDT,ITM)
      ENDIF

C
      ITM = ITM + 1
      CALL TIMINC(ETIME,ITMNEW,IINC)
      ETIME(1) = ITMNEW(1)
      ETIME(2) = ITMNEW(2)
      ETIME(3) = ITMNEW(3)

C
      IF (NINT(TCONV3(ETIME,ISTART)).LE.NINT(REND)) GO TO 20
C
      IF (I.EQ.KP) ITMMAX = ITM-1
      ENDIF

C
10    CONTINUE
C
      RETURN
      END

C
C ***** INDATA *****
C
      SUBROUTINE INDATA(PARAM,STARTT,ENDT,NDIM,DARRY,NUMNUM)
C
C  THIS IS A DUMMY SUBROUTINE TO READ IN DATA FOR THE
C  MAGNETOSHPERIC SPECIFICATION MODEL DURING DEVELOPMENT
C  AND TESTING.  THIS ROUTINE WILL BE REPLACED IN THE
C  OPERATIONAL MSM WITH AN INTERFACE ROUTINE TO THE
C  AWS ENVIRONMENTAL DATABASE.
C
C  BRYAN BALES          9/12/89
C
C  CHARACTER*(*) PARAM
C
C  DIMENSION STARTT(3),ENDT(3),DARRY(7,NDIM)
C
C  NUMNUM = 1
C  OPEN (UNIT=98,FILE=PARAM,STATUS='OLD')
C
10    CONTINUE
      READ (98,*,END=20) (DARRY(I,NUMNUM),I=1,7)
      NUMNUM = NUMNUM + 1
      IF (NUMNUM.LE.NDIM) GO TO 10
C
      STOP 'Stopping in INDATA. NUMNUM is greater than NDIM.'
C
20    CONTINUE
      NUMNUM = NUMNUM - 1
      CLOSE (98)
C
```

MSM56320
MSM56330
MSM56340
MSM56350
MSM56360
MSM56370
MSM56380
MSM56390
MSM56400
MSM56410
MSM56420
MSM56430
MSM56440
MSM56450
MSM56460
MSM56470
MSM56480
MSM56490
MSM56500
MSM56510
MSM56520
MSM56530
MSM56540
MSM56550
MSM56560
MSM56570
MSM56580
MSM56590
MSM56600
MSM56610
MSM56620
MSM56630
MSM56640
MSM56650
MSM56660
MSM56670
MSM56680
MSM56690
MSM56700
MSM56710
MSM56720
MSM56730
MSM56740
MSM56750
MSM56760
MSM56770
MSM56780
MSM56790
MSM56800
MSM56810
MSM56820
MSM56830
MSM56840
MSM56850
MSM56860
MSM56870
MSM56880
MSM56890
MSM56900
MSM56910
MSM56920
MSM56930
MSM56940
MSM56950

90/06/30
12:03:44

msml.for

90

RETURN
END

***** GETMAT *****

SUBROUTINE GETMAT(IWANT,LATDIM,LTDIM,BFPAR,WORK,BFLIM,BFEXST)

THIS SUBROUTINE FINDS THE MAGNETIC FIELD MATRICES NEEDED TO
INTERPOLATE ONE MATRIX THAT IS REPRESENTATIVE OF CURRENT
CONDITIONS.

BRYAN BALES 9/12/89

PARAMETER (IMSTND=5,IMTILT=5,IMINED=16,IMDST=8,IMSTCH=2)

INTEGER BFNDX(2,5)
LOGICAL*1 BFEXST(2,2,2,2,2)
LOGICAL MEXIST

DIMENSION STNDPR(IMSTND),TILTPR(IMTILT),FINEDP(IMINED)
DIMENSION DSTPR(IMDST),STCHPR(IMSTCH)
DIMENSION BFLIM(2,5),BFPAR(5)
DIMENSION WORK(LATDIM,LTDIM,2,2,2,2,2)
DIMENSION XDEB(62,51),YDEB(62,51),ZDEB(62,51)
DATA MSTND/IMSTND/
DATA MTILT/IMTILT/
DATA MINED/IMINED/
DATA MDST/IMDST/
DATA MSTCH/IMSTCH/

DATA STNDPR/6.0,8.0,10.0,12.0,14.0/
DATA TILTPR/-35.0,-17.5,0.0,17.5,35.0/
DATA FINEDP/49.47, 53.20, 55.89, 57.96, 59.60,
1 60.95, 62.06, 63.02, 63.99, 65.01,
2 65.91, 66.72, 67.45, 68.12, 68.73,
3 69.30/
DATA DSTPR/-400.0, -300.0, -200.0, -150.0,
1 -100.0, -50.0, 0.0, 50.0/
DATA STCHPR/0.0, 1.0/

EXTERNAL MEXIST

CALL FNDBRK(BFPAR(1),STNDPR,MSTND,BFNDX(1,1),BFNDX(2,1))
CALL FNDBRK(BFPAR(2),TILTPR,MTILT,BFNDX(1,2),BFNDX(2,2))
CALL FNDBRK(BFPAR(3),FINEDP,MINED,BFNDX(1,3),BFNDX(2,3))
CALL FNDBRK(BFPAR(4),DSTPR,MDST,BFNDX(1,4),BFNDX(2,4))
CALL FNDBRK(BFPAR(5),STCHPR,MSTCH,BFNDX(1,5),BFNDX(2,5))

DO 5, I=1,2
BFLIM(I,1) = STNDPR(BFNDX(I,1))
BFLIM(I,2) = TILTPR(BFNDX(I,2))
BFLIM(I,3) = FINEDP(BFNDX(I,3))
BFLIM(I,4) = DSTPR(BFNDX(I,4))
BFLIM(I,5) = STCHPR(BFNDX(I,5))

CONTINUE

DO 10, I=1,2
DO 20, J=1,2
DO 30, K=1,2
DO 40, L=1,2
DO 50, M=1,2
IF (MEXIST(BFNDX(I,1),BFNDX(J,2),BFNDX(K,3),
BFNDX(L,4),BFNDX(M,5))) THEN

MSM56960
MSM56970
MSM56980
MSM56990
MSM57000
MSM57010
MSM57020
MSM57030
MSM57040
MSM57050
MSM57060
MSM57070
MSM57080
MSM57090
MSM57100
MSM57110
MSM57120
MSM57130
MSM57140
MSM57150
MSM57160
MSM57170
MSM57180
MSM57190
MSM57200
MSM57210
MSM57220
MSM57230
MSM57240
MSM57250
MSM57260
MSM57270
MSM57280
MSM57290
MSM57300
MSM57310
MSM57320
MSM57330
MSM57340
MSM57350
MSM57360
MSM57370
MSM57380
MSM57390
MSM57400
MSM57410
MSM57420
MSM57430
MSM57440
MSM57450
MSM57460
MSM57470
MSM57480
MSM57490
MSM57500
MSM57510
MSM57520
MSM57530
MSM57540
MSM57550
MSM57560
MSM57570
MSM57580
MSM57590

90/06/30
12:03:44

91

mms1.for

```
1          CALL LOADBM(IWANT,LATDIM,LTDIM,I,J,K,L,M,BFNDX, MSM57600
              WORK)                                         MSM57610
          BFEKST(I,J,K,L,M) = .TRUE.                        MSM57620
ELSE                                              MSM57630
          CALL ZEROBM(LATDIM,LTDIM,I,J,K,L,M,WORK)        MSM57640
          BFEKST(I,J,K,L,M) = .FALSE.                     MSM57650
ENDIF                                           MSM57660
C                                              MSM57670
IF(BFNDX(J,2).EQ.3.AND.IWANT.EQ.1) THEN        MSM57680
  IF((3-BFNDX(M,5)).EQ.1.OR.BFPAR(5).LT.1) THEN MSM57690
    WRITE (6,*) 'STANDOFF =',BFLIM(I,1),' TILT =',      MSM57700
    BFLIM(J,2),' EQEDGE =',BFLIM(K,3),                MSM57710
    ' DST =',BFLIM(L,4),' COLLAPSE',                  MSM57720
    BFLIM(M,5)                                         MSM57730
    WRITE (6,1001) BFNDX(J,2),BFNDX(I,1),BFNDX(L,4), MSM57740
    BFNDX(K,3), (3-BFNDX(M,5))                        MSM57750
    FORMAT (' BF',I1,I1,I1,I2.2,I1)                  MSM57760
    WRITE(6,1001) BFNDX(J,2),BFNDX(I,1),BFNDX(L,4), MSM57770
    BFNDX(K,3), (3-BFNDX(M,5)),BFEKST(I,J,K,L,M),    MSM57780
    BFLIM(I,1),BFLIM(J,2),BFLIM(K,3),                MSM57790
    BFLIM(L,4),BFLIM(M,5)                            MSM57800
1001  FORMAT(' BF',I1,I1,I1,I2.2,I1,1X,L1,3X,'STANDOFF= ',F8.2,3X,
2      ' TILT= ',F8.2,3X,'EQEDGE= ',F8.2,3X,'DST= ',F8.2,3X,
3      ' COLLAPSE= ',F4.2)
    END IF
  END IF
C                                              MSM57810
C                                              MSM57820
C                                              MSM57830
C                                              MSM57840
C                                              MSM57850
C                                              MSM57860
C                                              MSM57870
C                                              MSM57880
C                                              MSM57890
C                                              MSM57900
998      CONTINUE                                         MSM57910
999      CONTINUE                                         MSM57920
C                                              MSM57930
C      1      CALL OUTP(XDEB,LATDIM,LTDIM,10,LATDIM,1,3, MSM57940
              LTDIM-3,1,1., 'XDEB',6,132)              MSM57950
C                                              MSM57960
C                                              MSM57970
C                                              MSM57980
C                                              MSM57990
C                                              MSM58000
996      CONTINUE                                         MSM58010
997      CONTINUE                                         MSM58020
C                                              MSM58030
C      1      CALL OUTP(YDEB,LATDIM,LTDIM,10,LATDIM,1,3, MSM58040
              LTDIM-3,1,1., 'YDEB',6,132)              MSM58050
C                                              MSM58060
C                                              MSM58070
C                                              MSM58080
C                                              MSM58090
994      CONTINUE                                         MSM58100
995      CONTINUE                                         MSM58110
C                                              MSM58120
C      1      CALL OUTP(ZDEB,LATDIM,LTDIM,10,LATDIM,1,3, MSM58130
              LTDIM-1,1,1., 'ZDEB',6,132)              MSM58140
C                                              MSM58150
C                                              MSM58160
C                                              MSM58170
C                                              MSM58180
C      10     CONTINUE                                     MSM58190
C                                              MSM58200
C                                              MSM58210
C              RETURN                                     MSM58220
C              END                                         MSM58230
```

90/06/30
12:03:44

msml.for

92

```
C ***** FND BRK *****MSM58240
C                                     MSM58250
C      SUBROUTINE FND BRK(PARVAL,PVALS,IPDIM,MIN,MAX)      MSM58260
C                                                         MSM58270
C      THIS SUBROUTINE FINDS THE INDICES OF THE VALUES IN ARRAY MSM58280
C      PVALS THAT BRACKET PARVAL.                        MSM58290
C                                                         MSM58300
C      BRYAN BALES                      9/12/89           MSM58310
C                                                         MSM58320
C      DIMENSION PVALS(IPDIM)                          MSM58330
C                                                         MSM58340
C      MIN = 0                                           MSM58350
C                                                         MSM58360
C      10 CONTINUE                                       MSM58370
C          MIN = MIN + 1                                  MSM58380
C          MAX = MIN + 1                                  MSM58390
C      IF (PARVAL.LE.PVALS(MIN)) THEN                    MSM58400
C          MAX = MIN                                       MSM58410
C      ELSEIF (PARVAL.GT.PVALS(MAX).AND.MAX.EQ.IPDIM) THEN MSM58420
C          MIN = MAX                                       MSM58430
C      IF (PARVAL.GT.PVALS(MAX).AND.MAX.LT.IPDIM) THEN   MSM58440
C          GO TO 10                                        MSM58450
C      ENDIF                                             MSM58460
C                                                         MSM58470
C      RETURN                                           MSM58480
C      END                                              MSM58490
C                                                         MSM58500
C ***** MEXIST *****MSM58510
C                                     MSM58520
C      LOGICAL FUNCTION MEXIST(MSTND,MTILT,MINED,MDST,MSTCH) MSM58530
C                                                         MSM58540
C      THIS FUNCTION CHECKS TO SEE IF A PARTICULAR B-MATRIX EXISTS MSM58550
C      IN THE B-SUPERMATRIX(TM).                        MSM58560
C                                                         MSM58570
C      CHARACTER*8 FILNAM                               MSM58580
C      LOGICAL TEST                                     MSM58590
C                                                         MSM58600
C      WRITE (FILNAM,1000) MTILT,MSTND,MDST,MINED,(3-MSTCH) MSM58610
C      1000 FORMAT ('BFIELD:BF',I1,I1,I1,I2.2,I1,'.DAT') MSM58620
C                                                         MSM58630
C      INQUIRE (FILE=FILNAM,EXIST=TEST)                 MSM58640
C      MEXIST = TEST                                     MSM58650
C                                                         MSM58660
C      RETURN                                           MSM58670
C      END                                              MSM58680
C                                                         MSM58690
C ***** ZERO BM *****MSM58700
C                                     MSM58710
C      SUBROUTINE ZERO BM(LATDIM,LTDIM,I,J,K,L,M,WORK)    MSM58720
C                                                         MSM58730
C      THIS SUBROUTINE ERASES AN INDIVIDUAL B-MATRIX WITHIN THE MSM58740
C      WORKING B-MATRICES. THIS IS DONE WHEN AN INDIVIDUAL MATRIX MSM58750
C      DOES NOT EXIST IN THE OFFLINE B-SUPERMATRIX.      MSM58760
C                                                         MSM58770
C      DIMENSION WORK(LATDIM,LTDIM,2,2,2,2)              MSM58780
C                                                         MSM58790
C      DO 10, LAT=1, LATDIM                               MSM58800
C          DO 20, LT=1, LTDIM                              MSM58810
C              WORK(LAT,LT,I,J,K,L,M) = 0.0              MSM58820
C          CONTINUE                                       MSM58830
C      10 CONTINUE                                       MSM58840
C                                                         MSM58850
C      RETURN                                           MSM58860
C      END                                              MSM58870
```

90/06/30
12:03:44

msml.for

93

```
C
C ***** LOADBM *****
C
C      SUBROUTINE LOADBM(IWANT,LATDIM,LTDIM,I,J,K,L,M,BFNDX,WORK)
C
C      THIS SUBROUTINE LOADS INDIVIDUAL B-MATRICES FROM THE OFFLINE
C      B-SUPERMATRIX INTO THE WORK B-MATRICES.  A CHECK IS MADE TO
C      VERIFY THAT THE CORRECT MATRIX HAS BEEN RETRIEVED.  IF THIS
C      CHECK FAILS THE PROGRAM WILL STOP.
C
C      INTEGER BFNDX(2,5)
C      CHARACTER*32 FILNAM
C
C      DIMENSION WORK(LATDIM,LTDIM,2,2,2,2,2)
C
C      THE FILE DEFINITIONS FOR THE B-MATRICES IN THE SUPERMATRIX
C      ARE AS FOLLOWS :
C          'BFvwxyyz'.
C          WHERE v IS THE TILT INDEX,
C                w IS THE STANDOFF INDEX,
C                x IS THE DST INDEX,
C                yy IS THE INNER EDGE INDEX,
C                z IS THE COLLAPSE INDEX.
C
C      WRITE (FILNAM,1000) BFNDX(J,2), BFNDX(I,1), BFNDX(L,4),
1      BFNDX(K,3), (3-BFNDX(M,5))
1000  FORMAT ('BFIELD:BF',I1,I1,I1,I2.2,I1,'.DAT')
C
C      OPEN (UNIT=99,FILE=FILNAM,ERR=199,FORM='FORMATTED',
1      ACCESS='SEQUENTIAL',STATUS='OLD')
C
C      READ (99,800) IDF1, IDF2, IDF3, IDF4, IDF5, IDF6
800  FORMAT(3I1,I2.2,I1,I3)
C
C      IF ((BFNDX(I,1).NE.IDF2).OR.(BFNDX(J,2).NE.IDF1).OR.
1      (BFNDX(K,3).NE.IDF4).OR.(BFNDX(L,4).NE.IDF3).OR.
2      ((3-BFNDX(M,5)).NE.IDF5).OR.(IDF6.NE.201)) THEN
      WRITE (LUERR,*) 'FILE ',FILNAM,' IS INCORRECT.'
      WRITE (6,*) 'FILE ',FILNAM,' IS INCORRECT.'
      STOP 'INCORRECT B-FIELD. STOPPING IN LOADBM.'
      ENDIF
C
C      DO 10, N=1,IWANT
      READ(99,801)((WORK(II,JJ,I,J,K,L,M),II=1,LATDIM),JJ=1,LTDIM)
801  FORMAT(11E12.4)
10  CONTINUE
C
C      CLOSE (99)
C      RETURN
C
199  CONTINUE
      WRITE(6,*) 'STOPPING IN LOADBM. OPEN FAILED ON ',FILNAM
      STOP 'STOPPING IN LOADBM. OPEN FAILED ON B-MATRIX.'
      END
C
C ***** TCONV2 *****
C
C      REAL FUNCTION TCONV2(YR,DD,STRTYR,STRTDD)
C
C      THIS FUNCTION TAKES A START YEAR, START DECIMAL DAY,
C      A CURRENT YEAR, AND A CURRENT DECIMAL DAY.  IT PRODUCES
C      A VALUE THAT IS THE SECONDS FROM THE START YEAR AND DAY.
C      THIS VALUE IS RELATIVE TO MIDNIGHT OF THE START DAY.
C
```

90/06/30
12:03:44

msml.for

94

INTRINSIC INT

MSM59520

MSM59530

MSM59540

MSM59550

MSM59560

MSM59570

MSM59580

MSM59590

MSM59600

MSM59610

MSM59620

MSM59630

MSM59640

MSM59650

MSM59660

MSM59670

MSM59680

MSM59690

MSM59700

MSM59710

MSM59720

MSM59730

MSM59740

MSM59750

MSM59760

MSM59770

MSM59780

MSM59790

MSM59800

MSM59810

MSM59820

MSM59830

MSM59840

MSM59850

MSM59860

MSM59870

MSM59880

MSM59890

MSM59900

MSM59910

MSM59920

MSM59930

MSM59940

MSM59950

MSM59960

MSM59970

MSM59980

MSM59990

MSM60000

MSM60010

MSM60020

MSM60030

MSM60040

MSM60050

MSM60060

MSM60070

MSM60080

MSM60090

MSM60100

MSM60110

MSM60120

MSM60130

MSM60140

MSM60150

TEMP = 0.0

IF (STRTYR.EQ.YR) THEN

TEMP = (DD - INT(STRTDD)) * 86400.0

ELSEIF (MOD(INT(STRTYR),4).EQ.0) THEN

TEMP = (DD + 366.0 - INT(STRTDD)) * 86400.0

ELSE

TEMP = (DD + 365.0 - INT(STRTDD)) * 86400.0

ENDIF

TCONV2 = TEMP

RETURN

END

***** TCONV3 *****

REAL FUNCTION TCONV3(ITIME,ISTART)

THIS FUNCTION TAKES A START YEAR, START DAY, A CURRENT YEAR,
A CURRENT DAY, AND A CURRENT SECOND.. IT PRODUCES
A VALUE THAT IS THE SECONDS FROM THE START YEAR AND DAY.
THIS VALUE IS RELATIVE TO MIDNIGHT OF THE START DAY.

PARAMETER (IYR=1,IDY=2,ISEC=3)

DIMENSION ITIME(3),ISTART(3)

INTRINSIC REAL

IF (ISTART(IYR).EQ.ITIME(IYR)) THEN

TEMP = REAL(ITIME(IDY) - ISTART(IDY))*86400.0 +

1 REAL(ITIME(ISEC))

ELSEIF (MOD(ISTART(IYR),4).EQ.0) THEN

TEMP = REAL(ITIME(IDY) + 366 - ISTART(IDY))*86400.0 +

1 REAL(ITIME(ISEC))

ELSE

TEMP = REAL(ITIME(IDY) + 365 - ISTART(IDY))*86400.0 +

1 REAL(ITIME(ISEC))

ENDIF

TCONV3 = TEMP

RETURN

END

***** SMOOTH *****

SUBROUTINE SMOOTH(DARRY,NDIM,NUMNUM,STARTY,STARTD,NA,XA,YA,IERR)

DIMENSION DARRY(7,NDIM),XA(NDIM),YA(NDIM)

REAL STARTY,STARTD

INTEGER NUMNUM,NDIM,NA,IERR

EXTERNAL TCONV2

This routine checks and extracts data from DARRY() into XA(),YA().
September 10, 1989 Akira NAGAI

INPUTS:

DARRY() : Source data

NDIM : A Dimension of DARRY()

NUMNUM : Number of source data

STARTY, STARTD : Start year and day

OUTPUTS:

C	NA,XA(),YA()	: # of valid data and valid data pairs	MSM60160
C	IERR	: ERROR CODE	MSM60170
C			MSM60180
C	September 10, 1989		MSM60190
C			MSM60200
C	Time conversion		MSM60210
C	DO 5,I=1,NUMNUM		MSM60220
C	XA(I) = TCONV2(DARRY(2,I),DARRY(3,I),STARTY,STARTD)		MSM60230
C	WRITE(6,*) 'SMOOTH --',DARRY(2,I),DARRY(3,I),STARTD,XA(I)		MSM60240
	YA(I) = DARRY(1,I)		MSM60250
5	CONTINUE		MSM60260
	NA = NUMNUM		MSM60270
			MSM60280
C	Check Time-tag. Shift-up if same.		MSM60290
	IERR=0		MSM60300
	I=0		MSM60310
10	CONTINUE		MSM60320
	I=I+1		MSM60330
	IF (I.GE.NA) RETURN		MSM60340
	IF (XA(I).GT.XA(I+1)) THEN		MSM60350
	WRITE(6,*) 'SERIOUS ERROR: TIME-TAG REVERSE ORDER IN DARRY()'		MSM60360
	WRITE(6,*) 'X(I) = ',XA(I),' X(I+1) = ',XA(I+1),' I = ',I		MSM60370
	IERR = -1		MSM60380
	RETURN		MSM60390
	ELSE		MSM60400
	IF (XA(I).EQ.XA(I+1)) THEN		MSM60410
	WRITE(6,*) 'WARNING: SAME TIME-TAG EXISTS IN DARRY()'		MSM60420
	WRITE(6,*) 'SECOND DATA IGNORED.'		MSM60430
	DO 20,K=I+1,NA-1		MSM60440
	XA(K)=XA(K+1)		MSM60450
	YA(K)=YA(K+1)		MSM60460
20	CONTINUE		MSM60470
	NA = NA - 1		MSM60480
	I = I - 1		MSM60490
	END IF		MSM60500
	END IF		MSM60510
	GOTO 10		MSM60520
	END		MSM60530
C			MSM60540
C	***** DTNTRP *****		MSM60550
C			MSM60560
	SUBROUTINE DTNTRP(X,NA,XA,YA,Y,DYDX,DELTA,IERR)		MSM60570
	INTEGER NA,IERR,I		MSM60580
	REAL X,XA(*),YA(*),XD,Y,DYDX,DELTA		MSM60590
C			MSM60600
C	This function returns interpolation values for the		MSM60610
C	N data pairs of (XA(i),YA(i)), where i=1..NA.		MSM60620
C			MSM60630
C	INPUTS:		MSM60640
C	X : Given value of X		MSM60650
C	NA : Number of data points.		MSM60660
C	XA : Data XA(1) to XA(NA)		MSM60670
C	YA : Data YA(1) to YA(NA)		MSM60680
C			MSM60690
C	OUTPUTS:		MSM60700
C	Y : Interpolated value of Y.		MSM60710
C	DYDX: Interpolated value of dY/dX.		MSM60720
C	DELTA : Gap of X.		MSM60730
C	IERR: Error code.		MSM60740
C			MSM60750
C	September 08, 1989		MSM60760
C			MSM60770
	IF (X.GE.XA(NA)) THEN		MSM60780
	IERR=-1		MSM60790

90/06/30
12:03:44

msm1.for

96

```
IF (X.EQ.XA(NA)) IERR=0
Y = YA(NA)
DYDX = ( YA(NA)-YA(NA-1) ) / ( XA(NA)-XA(NA-1) )
DELTA = X - XA(NA)
ELSE IF (X.LT.XA(1)) THEN
IERR=-1
Y = YA(1)
DYDX = ( YA(2)-YA(1) ) / ( XA(2)-XA(1) )
DELTA = XA(1) - X
ELSE
```

```
C Scan to find the position of X.
C Maybe binary search would be better.
```

```
10 I = NA
CONTINUE
XD = X - XA(I)
IF (XD.GT.0.0) GOTO 20
I = I - 1
GOTO 10
```

```
20 CONTINUE
IERR = 0
DELTA2= XA(I+1)-XA(I)
DELTA = MIN(X-XA(I),XA(I+1)-X)
DYDX = (YA(I+1)-YA(I)) / DELTA2
Y = DYDX * XD + YA(I)
ENDIF
RETURN
END
```

```
C ***** DTXIPT *****
```

```
C SUBROUTINE DTXIPT(X,NA,XA,YA,Y,DYDX,DELTA,IERR)
```

```
C INTEGER NA,IERR,I
C REAL X,XA(*),YA(*),XD,Y,DYDX,DELTA
```

```
C This function returns interpolation values for XIPATT
```

```
C INPUTS:
```

```
C X : Given value of X
C NA : Number of data points.
C XA : Data XA(1) to XA(NA)
C YA : Data YA(1) to YA(NA)
```

```
C OUTPUTS:
```

```
C Y : Interpolated value of Y.
C DYDX: Interpolated value of dY/dX.
C DELTA : Gap of X.
C IERR: Error code.
```

```
C September 08, 1989
```

```
C IF (X.GT.XA(NA)) THEN
IERR=-1
IF (X.EQ.XA(NA)) IERR=0
Y = YA(NA)
DELTA = X - XA(NA)
ELSE IF (X.LE.XA(1)) THEN
IERR=-1
IF (X.EQ.XA(1)) IERR=0
Y = YA(1)
DELTA = XA(1) - X
ELSE
```

MSM60800
MSM60810
MSM60820
MSM60830
MSM60840
MSM60850
MSM60860
MSM60870
MSM60880
MSM60890
MSM60900
MSM60910
MSM60920
MSM60930
MSM60940
MSM60950
MSM60960
MSM60970
MSM60980
MSM60990
MSM61000
MSM61010
MSM61020
MSM61030
MSM61040
MSM61050
MSM61060
MSM61070
MSM61080
MSM61090
MSM61100
MSM61110
MSM61120
MSM61130
MSM61140
MSM61150
MSM61160
MSM61170
MSM61180
MSM61190
MSM61200
MSM61210
MSM61220
MSM61230
MSM61240
MSM61250
MSM61260
MSM61270
MSM61280
MSM61290
MSM61300
MSM61310
MSM61320
MSM61330
MSM61340
MSM61350
MSM61360
MSM61370
MSM61380
MSM61390
MSM61400
MSM61410
MSM61420
MSM61430

90/06/30
12:03:44

msml.for

97

```
C      Scan to find the position of X.
      I = NA
10     CONTINUE
      XD = X - XA(I)
      IF (XD.GE.0.0) GOTO 20
      I = I - 1
      GOTO 10

20     CONTINUE
      IERR = 0
      DELTA = MIN(X-XA(I),XA(I+1)-X)
      A=X-XA(I)
      B=XA(I+1)-X

      IF (A.LT.B) THEN
        Y = YA(I)
      ELSE
        Y = YA(I+1)
      END IF
      ENDIF

      RETURN
      END

C*****STDOFF*****
C
C      SUBROUTINE STDOF(VELOC,DENS,STAND)
C
C          ANALYTIC CALCULATION OF THE STAND-OFF DISTANCE *
C
C          FORMULA ACCORD. TO ALPBACH - PAPER (1979)
C
C          B0 = MAGNETIC FIELD AT THE EQUATOR IN (GAUSS)
C          FAK = REFLECTION FACTOR FOR SOLAR-WIND PATICLES
C          F = GEOMETRY FAKTOR OF THE MAGNETOPAUSE
C          PSOL = SOLARWIND-PRESS IN (1.E-08 DYN/(CM**2))
C          PEFF = SOLARWIND-PRESS IN (1.E-16 DYN/(CM**2))
C
C          * * * INPUT PARAMTERS:      VELOC  SOL WIND VELOCITY IN [KM/SEC]
C          * * *                       DENS   SOL WIND DESITY IN [1/CM]**3
C
C          * * * OUTPUT PARAMETERS:    BSTAG  MAGNETIC FIELD AT SUBSOLAR POINT IN
C          * * *                       STAND  STAND OFF DISTANCE IN [EARTH RADII]
C
C          * * * CALCULATION OF HELIUM PERCENTAGE IN THE SOLAR WIND * * * * *
C
C          HELIUM = HELIUM-PROZENT
C
C          HELIUM = 0.00
C
C          PERC1 = HELIUM/100.
C          PERC2 = 1.0 - PERC1
C
C          DENGES = DENS * (PERC2 + 4.*PERC1)
C
C          V0 = VELOC
C
C          FAK = 1.0
C          PROTMA = 1.6726 * 1.E-24
C          VAU = V0 * 1.E+05
C          PI = ACOS(-1.00000)
```

MSM61440
MSM61450
MSM61460
MSM61470
MSM61480
MSM61490
MSM61500
MSM61510
MSM61520
MSM61530
MSM61540
MSM61550
MSM61560
MSM61570
MSM61580
MSM61590
MSM61600
MSM61610
MSM61620
MSM61630
MSM61640
MSM61650
MSM61660
MSM61670
MSM61680
MSM61690
MSM61700
MSM61710
MSM61720
MSM61730
MSM61740
MSM61750
MSM61760
MSM61770
MSM61780
MSM61790
MSM61800
MSM61810
MSM61820
MSM61830
MSM61840
MSM61850
MSM61860
MSM61870
MSM61880
MSM61890
MSM61900
MSM61910
MSM61920
MSM61930
MSM61940
MSM61950
MSM61960
MSM61970
MSM61980
MSM61990
MSM62000
MSM62010
MSM62020
MSM62030
MSM62040
MSM62050
MSM62060
MSM62070

90/06/30
12:03:44

msml.for

98

DRUCK = 8.*PI * FAK * DENGES * PROTMA * VAU * VAU
DRUCK = DRUCK * 1.E+10
DRUCK = SQRT(DRUCK)

FELD = DRUCK

PSOL = DRUCK * 1.E-05
PSOL = PSOL * PSOL
PSOL = PSOL / (8.*PI)
PSOL = PSOL * 1.E+08

PI = ACOS(-1.000000)
B0 = 0.23
B0 = 0.311829

PGES = PSOL * (PERC2 + 4.*PERC1)

* * IN THIS SPECIAL VERSION OF THE ROUTINE STDOFF, THE MAGNETOSPHERE
ASSUMED TO BE COMPLETELY CLOSED.

BZIMF = 0.00
CD = 0.00
CIMF = 0.00

BZIMF = BZIMF * 1.E-05

F = 1.16000
FAK = 1.0
FQK = F * F / FAK

PEFF = 2.*PI * PGES * 1.E-08
EX = 1./6.

R0 = (FQK * B0*B0*(1.-CD)*(1.-CD) / PEFF)**EX
R0 = R0 * (1. + (EX*CIMF*BZIMF) / (SQRT(PEFF)))

FELD = PGES * 1.E-08
FELD = FELD * 8.*PI
FELD = SQRT(FELD)
FELD = FELD * 1.E+05

BSTAG = FELD
STAND = R0

RETURN
END

SUBROUTINE FLXNIT(IRDIM, NRGDIM, KPDIM, KPPLUS, ISPDIM, FLXR,
2 FLXNRG, FLXKP, FLXMAT, THRMAT)

VERSION 1.00

DATE: 10.04.89

PURPOSE: SUBROUTINE TO CALCULATE KP-BASED PARTICLE NUMBER FLUX
IN UNITS OF PARTICLES/CM**2/S/SR/EV AS FCN OF R, ENERGY,
AND KP.

DIMENSION FLXMAT(IRDIM, NRGDIM, KPPLUS, ISPDIM)
DIMENSION THRMAT(IRDIM, NRGDIM, KPDIM)
DIMENSION FLXR(IRDIM), FLXNRG(NRGDIM), FLXKP(KPDIM)
DIMENSION THR(4)

MSM62080
MSM62090
MSM62100
MSM62110
MSM62120
MSM62130
MSM62140
MSM62150
MSM62160
MSM62170
MSM62180
MSM62190
MSM62200
MSM62210
MSM62220
MSM62230
MSM62240
MSM62250
MSM62260
MSM62270
MSM62280
MSM62290
MSM62300
MSM62310
MSM62320
MSM62330
MSM62340
MSM62350
MSM62360
MSM62370
MSM62380
MSM62390
MSM62400
MSM62410
MSM62420
MSM62430
MSM62440
MSM62450
MSM62460
MSM62470
MSM62480
MSM62490
MSM62500
MSM62510
MSM62520
MSM62530
MSM62540
MSM62550
MSM62560
MSM62570
MSM62580
MSM62590
MSM62600
MSM62610
MSM62620
MSM62630
MSM62640
MSM62650
MSM62660
MSM62670
MSM62680
MSM62690
MSM62700
MSM62710

90/06/30
12:03:44

msm1.for

99

```
C
DO 10 IKP=1,KPDIM
  FLXKP (IKP) =FLOAT (IKP-1)
10 CONTINUE
C
DO 20 INRG=1,NRGDIM
  FLXNRG (INRG) =FLOAT (INRG+3) /4.
20 CONTINUE
C
FLXR (1) =ALOG10 (3.)
FLXR (2) =ALOG10 (4.)
FLXR (3) =ALOG10 (6.6)
FLXR (4) =ALOG10 (13.)
C
C
DO 90 ISP=1,ISPDIM
  ISPP=ISP
  DO 100 IKP=1,KPDIM
    DO 110 INRG=1,NRGDIM
      CALL FLXCAL (ISPDIM,KPDIM,ISPP,FLXKP (IKP),FLXNRG (INRG),
2        FLX3,FLX4,FLX6,FLX13)
      DO 120 IR=1,IRDIM
        FLXMAT (IR,INRG,IKP,ISP) =FLXTRP (FLXR (IR),FLX3,FLX4,
2        FLX6,FLX13)
120      CONTINUE
110      CONTINUE
100      CONTINUE
90      CONTINUE
C
DO 200 IKP=1,KPDIM
  IIKP=IKP
  DO 210 INRG=1,NRGDIM
    IINRG=INRG
    CALL THRCAL (FLXKP (IKP),FLXNRG (INRG),IINRG,IIKP,IRDIM,
2    NRGDIM,KPDIM,KPPLUS,ISPDIM,FLXMAT,THR)
    DO 220 IR=1,IRDIM
      THRMAT (IR,INRG,IKP) =THR (IR)
220      CONTINUE
210      CONTINUE
200      CONTINUE
C
C CYCLE THROUGH KP VALUES AND PUT MIN FLUX IN KPDIM+1
C      MAX FLUX IN KPDIM+2
C
DO 300 ISP=1,ISPDIM
  DO 310 INRG=1,NRGDIM
    DO 320 IR=1,IRDIM
      FLXMIN=1.E20
      FLXMAX=-1.E20
      DO 330 IKP=1,KPDIM
        FLXMIN=MIN (FLXMAT (IR,INRG,IKP,ISP),FLXMIN)
        FLXMAX=MAX (FLXMAT (IR,INRG,IKP,ISP),FLXMAX)
330      CONTINUE
C
      FLXMAT (IR,INRG,KPDIM+1,ISP) =FLXMIN
      FLXMAT (IR,INRG,KPDIM+2,ISP) =FLXMAX
320      CONTINUE
310      CONTINUE
300      CONTINUE
C
C
C
RETURN
END
```

MSM62720
MSM62730
MSM62740
MSM62750
MSM62760
MSM62770
MSM62780
MSM62790
MSM62800
MSM62810
MSM62820
MSM62830
MSM62840
MSM62850
MSM62860
MSM62870
MSM62880
MSM62890
MSM62900
MSM62910
MSM62920
MSM62930
MSM62940
MSM62950
MSM62960
MSM62970
MSM62980
MSM62990
MSM63000
MSM63010
MSM63020
MSM63030
MSM63040
MSM63050
MSM63060
MSM63070
MSM63080
MSM63090
MSM63100
MSM63110
MSM63120
MSM63130
MSM63140
MSM63150
MSM63160
MSM63170
MSM63180
MSM63190
MSM63200
MSM63210
MSM63220
MSM63230
MSM63240
MSM63250
MSM63260
MSM63270
MSM63280
MSM63290
MSM63300
MSM63310
MSM63320
MSM63330
MSM63340
MSM63350

90/06/30
12:03:44

msm1.for

100

SUBROUTINE FLXCAL (ISPDIM, KPDIM, ISPP, FKP, FNRGLG, FLX3, FLX4, FLX6,
2 FLX13)

VERSION 1.00

DATE: 05.02.90

PURPOSE: COMPUTE KP-DEPENDENT FLUX (PARTICLES/CM**2/S/SR/EV) AT
L=3, 4, 6.6, AND 13 FOR ELECTRONS OR IONS OF ENERGY
FNRG (EV) AND FOR KP=FKP CONDITIONS. ISPP GIVES THE
MASS SPECIES OF THE FLUX TO BE CALCULATED.
IF ISPP=1 THEN ELECTRON FLUX VALUES ARE RETURNED
IF ISPP=2 THEN H+ FLUX VALUES ARE RETURNED
IF ISPP=3 THEN O+ FLUX VALUES ARE RETURNED

REF: SECTION 2.5 OF THE FINAL REPORT FOR
CONTRACT #F19628-87-K-0001

PARAMETER (ISPSIZ=3, IKPSIZ=7, KAPSIZ=2)
PARAMETER (NKPSIZ=3, NSPSIZ=2)

DIMENSION DENGEO (KAPSIZ, IKPSIZ), EGEO (KAPSIZ, IKPSIZ)
DIMENSION DIONG (KAPSIZ, IKPSIZ), EIONG (KAPSIZ, IKPSIZ)

DIMENSION DEN4 (NKPSIZ, KAPSIZ, NSPSIZ), DEN3 (NKPSIZ, KAPSIZ, NSPSIZ)
DIMENSION EZ43 (NKPSIZ, KAPSIZ, NSPSIZ), RKAP43 (NKPSIZ, KAPSIZ, NSPSIZ)
DIMENSION FKAP43 (NKPSIZ, KAPSIZ, NSPSIZ), FLX43 (NKPSIZ)

DATA DENGEO /0.5697, .004926, 0.6034, .2261, 0.6292, .2583,
2 0.7002, .3418, 0.8135, .4602, 1.0347, .6446,
3 1.4468, .8505/

DATA EGEO /31.036, 21000., 40.803, 4832.4, 48.670, 4814.0,
2 71.680, 4797.8, 111.119, 4816.0, 192.704, 4894.3,
3 342.712, 5075.2/

DATA DIONG /0.602, 0.460, 0.661, 0.457,
2 0.712, 0.495, 0.813, 0.524,
3 0.966, 0.539, 0.700, 0.517,
4 1.652, 0.453/

DATA EIONG / 180.2, 25408., 258.4, 26837.,
2 335.4, 27902., 604.8, 30723.,
3 1144.2, 34972., 2315.4, 42809.,
4 4182.4, 57163./

DATA DEN4 /0.10, 0.20, 0.00, 0.375, 1.90, 2.30,
2 0.43, 0.18, 0.00, 0.00, 1.50, 5.00/

DATA DEN3 /0.032, 0.06, 0.00, 0.12, 0.60, 6.30,
2 0.06, 0.018, 0.0, 0.00, 0.15, 11.8/

DATA EZ43 /3.00, 15.0, 15.00, 200., 40.0, 60.0,
2 5.00, 5.00, 5.00, 20.0, 20.0, 30.0/

DATA FKAP43 / .9359421, .9619483, .9619483,
2 .9619483, .9359421, .9619483,
3 .9027036, .9619483, .9619483,
4 .9359421, .9359421, .9619483/

DATA RKAP43 /6.00, 10.0, 10.00, 10.0, 6.00, 10.0,
2 4.00, 10.0, 10.00, 6.00, 6.00, 10.0/

MSM63360
MSM63370
MSM63380
MSM63390
MSM63400
MSM63410
MSM63420
MSM63430
MSM63440
MSM63450
MSM63460
MSM63470
MSM63480
MSM63490
MSM63500
MSM63510
MSM63520
MSM63530
MSM63540
MSM63550
MSM63560
MSM63570
MSM63580
MSM63590
MSM63600
MSM63610
MSM63620
MSM63630
MSM63640
MSM63650
MSM63660
MSM63670
MSM63680
MSM63690
MSM63700
MSM63710
MSM63720
MSM63730
MSM63740
MSM63750
MSM63760
MSM63770
MSM63780
MSM63790
MSM63800
MSM63810
MSM63820
MSM63830
MSM63840
MSM63850
MSM63860
MSM63870
MSM63880
MSM63890
MSM63900
MSM63910
MSM63920
MSM63930
MSM63940
MSM63950
MSM63960
MSM63970
MSM63980
MSM63990

90/06/30
12:03:44

msml.for

101

```
IF (ISPSIZ.NE.ISPDIM) THEN
  WRITE(6,*) 'DIMENSION ERROR IN SUBROUTINE FLXNIT'
  WRITE(6,*) 'ISPSIZ IS NOT EQUAL TO ISPDIM'
  WRITE(6,*) 'ISPSIZ=',ISPSIZ,'ISPDIM=',ISPDIM
  WRITE(6,*) 'STOPPING PROGRAM IN FLXNIT'
  STOP
END IF

C
IF (IKPSIZ.NE.KPDIM) THEN
  WRITE(6,*) 'DIMENSION ERROR IN SUBROUTINE FLXNIT'
  WRITE(6,*) 'IKPSIZ IS NOT EQUAL TO KPDIM'
  WRITE(6,*) 'IKPSIZ=',ISPSIZ,'KPDIM=',KPDIM
  WRITE(6,*) 'STOPPING PROGRAM IN FLXNIT'
  STOP
END IF

C
FNRG=10.**FNRGLG

C
IF (ISPP.EQ.1) THEN
  C
  CALCULATE ELECTRON FLUX VALUES
  RKAP=6.
  FKAP=.9359421

  C
  CALCULATE PLASMA SHEET FLUX, FLX13
  DENSITY=(.4*(FKP-1.)+0.5*(5.-FKP))/4.
  EZERO=.2885*(FKP-1.)+.15625*(5.-FKP)
  FLX13=1.68E5*FKAP*DENSITY*(FNRG*1.E-3/EZERO)/SQRT(EZERO)/
  2 (1.+(FNRG*1.E-3)/RKAP/EZERO)**(RKAP+1.)

  C
  C
  C
  CALCULATE GEOSYNCHRONOUS FLUX, FLX6
  IKP=NINT(FKP+1.)
  FLX6=0.
  DO 10 N=1,2
    C
    WRITE(6,*) 'IKP=',IKP,' N=',N,' FNRG=',FNRG
    C
    WRITE(6,*) 'DENSITY=',DENGEO(N,IKP),' EZERO=',EGEO(N,IKP)
    C
    WRITE(6,*) 'FKAP=',FKAP,' RKAP=',RKAP
    FLX6=FLX6+1.68E5*FKAP*DENGEO(N,IKP)*(FNRG/EGEO(N,IKP))/
    2 SQRT(EGEO(N,IKP)*1.E-3)/
    3 (1.+MIN(FNRG/RKAP/EGEO(N,IKP),1.E5))** (RKAP+1.)
    C
    WRITE(6,*) 'FLX6=',FLX6
  10 CONTINUE

  C
  C
  C
  CALCULATE FLUX AT L=4, FLX4
  FLX4=1.106E1*(FNRG*1.E-3)*(10.**((FKP-2.)/6.))/
  2 (1.+FNRG*1.E-3/32.))**3
  C
  WRITE(6,*) 'FLX4=',FLX4

  C
  C
  C
  CALCULATE FLUX AT L=3, FLX3
  FLX3=4.795E0*(FNRG*1.E-3)*(10.**((FKP-2.)/6.))/
  2 (1.+FNRG*1.E-3/65.8))**3.8

  C
  C
  C
  ELSE
    C
    CALCULATE H+ OR O+ FLUXES

    C
    C
    C
    PLASMA SHEET FLUX, FLX13
    DENSITY=(.4*(FKP-1.)+0.5*(5.-FKP))/4.
    EZERO=2.25*(FKP-1.)+1.219*(5.-FKP)
    C
    EZERO IN KEV
```

MSM64000
MSM64010
MSM64020
MSM64030
MSM64040
MSM64050
MSM64060
MSM64070
MSM64080
MSM64090
MSM64100
MSM64110
MSM64120
MSM64130
MSM64140
MSM64150
MSM64160
MSM64170
MSM64180
MSM64190
MSM64200
MSM64210
MSM64220
MSM64230
MSM64240
MSM64250
MSM64260
MSM64270
MSM64280
MSM64290
MSM64300
MSM64310
MSM64320
MSM64330
MSM64340
MSM64350
MSM64360
MSM64370
MSM64380
MSM64390
MSM64400
MSM64410
MSM64420
MSM64430
MSM64440
MSM64450
MSM64460
MSM64470
MSM64480
MSM64490
MSM64500
MSM64510
MSM64520
MSM64530
MSM64540
MSM64550
MSM64560
MSM64570
MSM64580
MSM64590
MSM64600
MSM64610
MSM64620
MSM64630

90/06/30
12:03:44

msml.for

102

```

RKAP=6.
FKAP=.9359421
C
IF (FKP.GE.1.) THEN
    RATO= (0.6*(FKP-1.)+0.06*(5.-FKP))/4.
ELSE
    RATO=0.06
END IF
C
IF (ISPP.EQ.2) THEN
    DENION=DENSTY/(1.+RATO)
    AWT=1.
ELSE
    DENION=DENSTY*RATO/(1.+RATO)
    AWT=16.
END IF
C
FLX13=3.93E3*FKAP*DENION*(FNRG*1.E-3/EZERO)/SQRT(AWT*EZERO)/
2      (1.+(FNRG*1.E-3)/RKAP/EZERO)**(RKAP+1.)
C
C
C CALCULATE GEOSYNCHRONOUS ION FLUX, FLX6
    RKAP1=10.
    RKAP2=4.
    FKAP1=.96194826
    FKAP2=.9027036
C
    RATO1=(0.7*(FKP-1.)+0.4*(5.-FKP))/4.
C
IF (FKP.GE.1.) THEN
    RATO2=(0.154*(5.-FKP)+0.767*(FKP-1.))/4.
ELSE
    RATO2=0.154
END IF
C
IKP=NINT(FKP+1.)
FLX6=0.
C
DO 20 N=1,2
C
    IF (N.EQ.1) THEN
        RKAP=RKAP1
        FKAP=FKAP1
        RATO=RATO1
    ELSE
        RKAP=RKAP2
        FKAP=FKAP2
        RATO=RATO2
    END IF
C
    IF (ISPP.EQ.2) THEN
        H+ IONS
        DENION=DIIONG(N,IKP)/(1.+RATO)
        AWT=1.
    ELSE
        O+ IONS
        DENION=DIIONG(N,IKP)*RATO/(1.+RATO)
        AWT=16.
    END IF
C
    FLX6=FLX6+3.93E3*FKAP*DENION*(FNRG/EIONG(N,IKP))/
2      SQRT(AWT*EIONG(N,IKP)*1.E-3)/
3      (1.+FNRG/RKAP/EIONG(N,IKP))** (RKAP+1.)
C
```

MSM64640
MSM64650
MSM64660
MSM64670
MSM64680
MSM64690
MSM64700
MSM64710
MSM64720
MSM64730
MSM64740
MSM64750
MSM64760
MSM64770
MSM64780
MSM64790
MSM64800
MSM64810
MSM64820
MSM64830
MSM64840
MSM64850
MSM64860
MSM64870
MSM64880
MSM64890
MSM64900
MSM64910
MSM64920
MSM64930
MSM64940
MSM64950
MSM64960
MSM64970
MSM64980
MSM64990
MSM65000
MSM65010
MSM65020
MSM65030
MSM65040
MSM65050
MSM65060
MSM65070
MSM65080
MSM65090
MSM65100
MSM65110
MSM65120
MSM65130
MSM65140
MSM65150
MSM65160
MSM65170
MSM65180
MSM65190
MSM65200
MSM65210
MSM65220
MSM65230
MSM65240
MSM65250
MSM65260
MSM65270

90/06/30
12:03:44

msm1.for

103

```
20      CONTINUE
C
C
C      CALCULATE ION FLUX AT L=4, FLX4
C      NOTE: THIS ALGORITHM IS A LITTLE WASTEFUL OF TIME IN THE INTEREST
C      OF SAVING PROGRAM COMPLEXITY.
C
C      ISPNDX=ISPP-1
C      DO 30 NKP=1,NKPSIZ
C
C      NKP=1 IS ASSOCIATED WITH KP=0
C      NKP=2 IS ASSOCIATED WITH KP=5+
C      NKP=3 IS ASSOCIATED WITH KP=9-
C
C      FLX43(NKP)=0.
C
C      DO 40 N=1,2
C      FLX43(NKP)=FLX43(NKP)+3.93E3*FKAP43(NKP,N,ISPNDX)*
2          DEN4(NKP,N,ISPNDX)*(FNRG*1.E-3/
3          EZ43(NKP,N,ISPNDX))/
4          Sqrt(AWT*EZ43(NKP,N,ISPNDX))/
5          (1.+(FNRG*1.E-3)/RKAP43(NKP,N,ISPNDX)/
6          EZ43(NKP,N,ISPNDX))** (RKAP43(NKP,N,ISPNDX)+1.)
C
40      CONTINUE
C
30      CONTINUE
C
C      IF (FKP.LE.5.34) THEN
C      FLXLOG=ALOG10(FLX43(1))+ALOG10(FLX43(2)/FLX43(1))*
2          (FKP/5.34)
C      ELSE IF (FKP.GT.5.34.AND.FKP.LE.8.67) THEN
C      FLXLOG=ALOG10(FLX43(2))+ALOG10(FLX43(3)/FLX43(2))*
2          ((FKP-5.34)/(8.67-5.34))
C      ELSE
C      FLXLOG=ALOG10(FLX43(3))
C      END IF
C
C      FLX4=10.**FLXLOG
C
C
C
C
C      CALCULATE ION FLUX AT L=3, FLX3
C      NOTE: THIS ALGORITHM IS A LITTLE WASTEFUL OF TIME IN THE INTEREST
C      OF SAVING PROGRAM COMPLEXITY.
C
C      ISPNDX=ISPP-1
C      DO 50 NKP=1,NKPSIZ
C
C      NKP=1 IS ASSOCIATED WITH KP=0
C      NKP=2 IS ASSOCIATED WITH KP=5+
C      NKP=3 IS ASSOCIATED WITH KP=9-
C
C      FLX43(NKP)=0.
C
C      DO 60 N=1,2
C      FLX43(NKP)=FLX43(NKP)+3.93E3*FKAP43(NKP,N,ISPNDX)*
2          DEN3(NKP,N,ISPNDX)*(FNRG*1.E-3/
3          EZ43(NKP,N,ISPNDX))/
4          Sqrt(AWT*EZ43(NKP,N,ISPNDX))/
5          (1.+(FNRG*1.E-3)/RKAP43(NKP,N,ISPNDX)/
6          EZ43(NKP,N,ISPNDX))** (RKAP43(NKP,N,ISPNDX)+1.)
C
```

MSM65280
MSM65290
MSM65300
MSM65310
MSM65320
MSM65330
MSM65340
MSM65350
MSM65360
MSM65370
MSM65380
MSM65390
MSM65400
MSM65410
MSM65420
MSM65430
MSM65440
MSM65450
MSM65460
MSM65470
MSM65480
MSM65490
MSM65500
MSM65510
MSM65520
MSM65530
MSM65540
MSM65550
MSM65560
MSM65570
MSM65580
MSM65590
MSM65600
MSM65610
MSM65620
MSM65630
MSM65640
MSM65650
MSM65660
MSM65670
MSM65680
MSM65690
MSM65700
MSM65710
MSM65720
MSM65730
MSM65740
MSM65750
MSM65760
MSM65770
MSM65780
MSM65790
MSM65800
MSM65810
MSM65820
MSM65830
MSM65840
MSM65850
MSM65860
MSM65870
MSM65880
MSM65890
MSM65900
MSM65910

20/06/30
12:03:44

msm1.for

104

```

60      CONTINUE
C
50      CONTINUE
C
      IF (FKP.LE.5.34) THEN
          FLXLOG=ALOG10 (FLX43 (1) )+ALOG10 (FLX43 (2) /FLX43 (1) ) *
2              (FKP/5.34)
      ELSE IF (FKP.GT.5.34.AND.FKP.LE.8.67) THEN
          FLXLOG=ALOG10 (FLX43 (2) )+ALOG10 (FLX43 (3) /FLX43 (2) ) *
2              ((FKP-5.34) / (8.67-5.34))
      ELSE
          FLXLOG=ALOG10 (FLX43 (3) )
      END IF
C
      FLX3=10.**FLXLOG
C
C
      END IF
C
      RETURN
      END
      FUNCTION FLXTRP (RLOG,FLX3,FLX4,FLX6,FLX13)
C
C  VERSION 1.00                      DATE: 10.04.89
C
C  PURPOSE: FUNCTION SUBPROGRAM TO RETURN FLUX VALUE INTERPOLATED
C            BETWEEN VALUES AT L=3, 4, 6.6, AND 13 RE.
C
C
      FR=10.**RLOG
      IF (FR.LE.4.) THEN
          FLXLOG=(ALOG10 (FR/3.) *ALOG10 (FLX4)+ALOG10 (4./FR) *ALOG10 (FLX3) )
2              /ALOG10 (4./3.)
      ELSE IF (FR.GT.4..AND.FR.LE.6.6) THEN
          FLXLOG=(ALOG10 (FR/4.) *ALOG10 (FLX6)+ALOG10 (6.6/FR) *ALOG10 (FLX4) )
2              /ALOG10 (6.6/4.)
      ELSE
          FLXLOG=ALOG10 (FLX13)+ALOG10 (13./FR)
      END IF
C
      FLXTRP=FLXLOG
C
C
      RETURN
      END
      FUNCTION FLXVAL (ISP,FKP,RR,PP,ENRG,IRDIM,NRGDIM,KPDIM,KPPLUS,
2              ISPDIM,FLXMAT,FLXR,FLXNRG,FLXKP)
C
C  VERSION 1.00                      DATE: 05.08.90
C                                      12.31.89
C                                      05.30.90
C
C  PURPOSE: CALCULATE KP-DEPENDENT FLUX VALUE AT R=RR, FOR PARTICLES OF
C            SPECIES ISP, OF ENERGY ENRG AND KP=FKP CONDITIONS BY
C            INTERPOLATING FLXMAT ARRAY.
C
C  PROGRAMMER: R.W. SPIRO
C
      DIMENSION FLXMAT (IRDIM,NRGDIM,KPPLUS,ISPDIM)
      DIMENSION FLXR (IRDIM) ,FLXNRG (NRGDIM) ,FLXKP (KPDIM)
C
C  DETERMINE INTERPOLATION INDICES
C
```

MSM65920
MSM65930
MSM65940
MSM65950
MSM65960
MSM65970
MSM65980
MSM65990
MSM66000
MSM66010
MSM66020
MSM66030
MSM66040
MSM66050
MSM66060
MSM66070
MSM66080
MSM66090
MSM66100
MSM66110
MSM66120
MSM66130
MSM66140
MSM66150
MSM66160
MSM66170
MSM66180
MSM66190
MSM66200
MSM66210
MSM66220
MSM66230
MSM66240
MSM66250
MSM66260
MSM66270
MSM66280
MSM66290
MSM66300
MSM66310
MSM66320
MSM66330
MSM66340
MSM66350
MSM66360
MSM66370
MSM66380
MSM66390
MSM66400
MSM66410
MSM66420
MSM66430
MSM66440
MSM66450
MSM66460
MSM66470
MSM66480
MSM66490
MSM66500
MSM66510
MSM66520
MSM66530
MSM66540
MSM66550


```

IF (FKP.GE.0.) THEN
  BKP=FKP+1.
  IF (BKP.GT.FLOAT(KPDIM)) BKP=FLOAT(KPDIM)
ELSE
  BKP=ABS (FKP)
END IF

BNRG=4.*ALOG10 (ENRG) -3.

RLOG=ALOG10 (RR)

IF (RLOG.GT.FLXR (IRDIM)) THEN
  BR=IRDIM
  FLXLOG=G3NTRP (FLXMAT (1,1,1,ISP), IRDIM,NRGDIM,KPPLUS,BR,BNRG,
2                                     BKP) + ALOG10 (13./RR)
ELSE
  DO 10 IR=1,IRDIM-1
    IF (RLOG.LE.FLXR (IR+1)) THEN
      BR=FLOAT (IR) + (RLOG-FLXR (IR)) / (FLXR (IR+1) -FLXR (IR))
      GO TO 20
    END IF
10  CONTINUE
20  CONTINUE
  WRITE (6,*) ' FKP,RR,ENRG',FKP,RR,ENRG
  WRITE (6,*) ' BKP,BR,BNRG',BKP,BR,BNRG

  FLXLOG=G3NTRP (FLXMAT (1,1,1,ISP), IRDIM,NRGDIM,KPPLUS,BR,BNRG,
2                                     BKP)
  END IF

  FLXVAL=10.**FLXLOG
  WRITE (6,*) '      FLXLOG=',FLXLOG,'      FLXVAL=',FLXVAL

  RETURN
END
SUBROUTINE EFIELD (KPATT,PCP,DEQDT,ASOUTH,BSOUTH,DXS,DYS,ANORTH,
2                BNORTH,DXN,DYN,COLAT,ALOC,MODE,LATDIM,LTDIM,
3                JWRAP,ITMDIM,ITMCUR,ICNTRL,ITOP,V,VNORTH,
4                VSOUTH)

C
C PURPOSE:  SUBROUTINE TO RETURN VALUES OF THE POTENTIAL ON THE
C           NORTHERN HEMISPHERE GRID (VNORTH), ON THE SOUTHERN
C           HEMISPHERE GRID (VSOUTH), AND THE AVERAGE OF THE TWO
C           HEMISPHERES (V).  THE POTENTIAL DISTRIBUTION GIVEN BY
C           V IS THE ACTUAL DISTRIBUTION USED TO FOLLOW PARTICLE
C           TRACES.  THIS ROUTINE CALLS SUBROUTINE EMODEL TO ACTUALLY
C           COMPUTE THE POTENTIAL DISTRIBUTIONS.
C
C NOTE:  KPATT GIVES THE UTD DEFINED PATTERN TYPE WITH 0 CORRESPONDING
C        TO NORTHWARD IMF CASES; 1 TO BZ SOUTH WITH STRONGER FLOW ON
C        THE DAWN SIDE OF THE POLAR CAP; 2 TO BZ SOUTH WITH SYMMETRIC
C        FLOW; AND 3 TO BZ SOUTH WITH STRONGER FLOW ON THE DUSK SIDE.
C        THIS SUBROUTINE CALLS EMODEL AFTER TRANSLATING THE PATTERN
C        TYPE TO THE INTERNAL EMODEL CONVENTION AS GIVEN BELOW:
C
C GENERAL CALL PARAMETERS FOR SUBROUTINE EMODEL
C
C        IPATT = HEPPNER-MAYNARD PATTERN NO.  MUST BE SPECIFIED AS AN
C               INTEGER BETWEEN 1 AND 7.
C               IPATT=1 MEANS PATTERN A, FOR BZ SOUTH, BY=0.

```

90/06/30
12:03:44

106

msml.for

```
C      IPATT=2 MEANS PATTERN BC, FOR BZ SOUTH, BYNORTH>0      MSM67200
C      IPATT=3 MEANS PATTERN DE, FOR BZ SOUTH, BYNORTH<0      MSM67210
C      IPATT=4 MEANS PATTERN BCP, TWISTED BC, FOR BZ>0        MSM67220
C      IPATT=5 MEANS PATTERN BCPP, TWISTED BC, BZ STRONG>0    MSM67230
C      IPATT=6 MEANS PATT. DEP, TWISTED DE, BZ WEAK >0       MSM67240
C      IPATT=7 MEANS PATT. DEPP, TWISTED DE, BZ STRONG>0      MSM67250
C                                                                MSM67260
C      FOR THE PRESENT, NORTHWARD IMF CASES ARE ALL TREATED AS IPATT=1  MSM67270
C                                                                MSM67280
C      PROGRAMMER:  R.W. SPIRO                                LAST UPDATE: 11-17-89  MSM67290
C                                                                MSM67300
C                                                                MSM67310
C      DIMENSION COLAT(LATDIM, LTDIM), ALOCT(LATDIM, LTDIM)    MSM67320
C      DIMENSION V(LATDIM, LTDIM, ITMDIM), VNORTH(LATDIM, LTDIM)  MSM67330
C      DIMENSION VSOUTH(LATDIM, LTDIM)                        MSM67340
C      DIMENSION ASOUTH(3, ITMDIM), BSOUTH(3, ITMDIM)         MSM67350
C      DIMENSION DXS(3, ITMDIM), DYS(3, ITMDIM)               MSM67360
C      DIMENSION ANORTH(3, ITMDIM), BNORTH(3, ITMDIM)         MSM67370
C      DIMENSION DXN(3, ITMDIM), DYN(3, ITMDIM)               MSM67380
C                                                                MSM67390
C                                                                MSM67400
C      IF (KPATT.EQ.0) THEN                                     MSM67410
C      NORTHWARD BZ CASE                                       MSM67420
C      IPATT=1                                                 MSM67430
C      CALL EMODEL(IPATT, PCP, DEQDT, ASOUTH(1, ITMCUR), BSOUTH(1, ITMCUR), MSM67440
C      2      DXS(1, ITMCUR), DYS(1, ITMCUR), COLAT, ALOCT,    MSM67450
C      3      MODE, LATDIM, LTDIM, JWRAP, ICNTRL, ITOP, VSOUTH) MSM67460
C      CALL EMODEL(IPATT, PCP, DEQDT, ANORTH(1, ITMCUR), BNORTH(1, ITMCUR), MSM67470
C      2      DXN(1, ITMCUR), DYN(1, ITMCUR), COLAT, ALOCT,    MSM67480
C      3      MODE, LATDIM, LTDIM, JWRAP, ICNTRL, ITOP, VNORTH) MSM67490
C                                                                MSM67500
C      ELSE IF (KPATT.EQ.-1.OR.KPATT.EQ.3) THEN               MSM67510
C      CASE OF BZ SOUTH, BY<0                                   MSM67520
C                                                                MSM67530
C      IPATT=2                                                 MSM67540
C      CALL EMODEL(IPATT, PCP, DEQDT, ASOUTH(1, ITMCUR), BSOUTH(1, ITMCUR), MSM67550
C      2      DXS(1, ITMCUR), DYS(1, ITMCUR), COLAT, ALOCT,    MSM67560
C      3      MODE, LATDIM, LTDIM, JWRAP, ICNTRL, ITOP, VSOUTH) MSM67570
C                                                                MSM67580
C      IPATT=3                                                 MSM67590
C      CALL EMODEL(IPATT, PCP, DEQDT, ANORTH(1, ITMCUR), BNORTH(1, ITMCUR), MSM67600
C      2      DXN(1, ITMCUR), DYN(1, ITMCUR), COLAT, ALOCT,    MSM67610
C      3      MODE, LATDIM, LTDIM, JWRAP, ICNTRL, ITOP, VNORTH) MSM67620
C                                                                MSM67630
C      ELSE IF (KPATT.EQ.1.OR.KPATT.EQ.-3) THEN               MSM67640
C      CASE OF BZ SOUTH, BY>0                                   MSM67650
C                                                                MSM67660
C      IPATT=3                                                 MSM67670
C      CALL EMODEL(IPATT, PCP, DEQDT, ASOUTH(1, ITMCUR), BSOUTH(1, ITMCUR), MSM67680
C      2      DXS(1, ITMCUR), DYS(1, ITMCUR), COLAT, ALOCT,    MSM67690
C      3      MODE, LATDIM, LTDIM, JWRAP, ICNTRL, ITOP, VSOUTH) MSM67700
C                                                                MSM67710
C      IPATT=2                                                 MSM67720
C      CALL EMODEL(IPATT, PCP, DEQDT, ANORTH(1, ITMCUR), BNORTH(1, ITMCUR), MSM67730
C      2      DXN(1, ITMCUR), DYN(1, ITMCUR), COLAT, ALOCT,    MSM67740
C      3      MODE, LATDIM, LTDIM, JWRAP, ICNTRL, ITOP, VNORTH) MSM67750
C                                                                MSM67760
C      ELSE                                                    MSM67770
C      CASE OF BZ SOUTH, SYMMETRIC FLOW                        MSM67780
C                                                                MSM67790
C      IPATT=1                                                 MSM67800
C      CALL EMODEL(IPATT, PCP, DEQDT, ASOUTH(1, ITMCUR), BSOUTH(1, ITMCUR), MSM67810
C      2      DXS(1, ITMCUR), DYS(1, ITMCUR), COLAT, ALOCT,    MSM67820
C      3      MODE, LATDIM, LTDIM, JWRAP, ICNTRL, ITOP, VSOUTH) MSM67830
```

90/06/30
12:03:44

msm1.for

107

```
C
      CALL EMODEL(IPATT,PCP,DEQDT,ANORTH(1,ITMCUR),BNORTH(1,ITMCUR),
2         DXN(1,ITMCUR),DYN(1,ITMCUR),COLAT,ALOCT,
3         MODE,LATDIM,LTDIM,JWRAP,ICNTRL,ITOP,VNORTH)
C
      END IF
C
C   COMPUTE V=POTENTIAL USED IN TRACING PARTICLE TRAJECTORIES=AVG OF
C       VNORTH AND VSOUTH
C
      DO 100 J=1,LTDIM
      DO 110 I=1,LATDIM
        V(I,J,ITMCUR)=(VSOUTH(I,J)+VNORTH(I,J))/2.
C
110    CONTINUE
100    CONTINUE
C
      RETURN
      END
      SUBROUTINE PWRCAL(LATDIM,LTDIM,ISPDIM,ITMDIM,ITMCUR,FKP,FLXSUM,
2         EAVG,COLAT,ALOCT,ALPHA,BETA,BNDLOC,A,B,DX,DY)
C
C   PURPOSE:  ESTIMATE PRECIPITATING ELECTRON ENERGY FLUX POLEWARD OF
C       THE MAIN MSM MODELING REGION BY COMPARING FLUX WITHIN
C       MODELING REGION WITH HARDY STATISTICAL VALUES.  THIS ROUTINE
C       IS BASED ON SUBROUTINE PWRCAL IN THE RCM SOURCE CODE; THE
C       RATIONALE FOR THE METHOD IS GIVEN IN AN R.A. WOLF DOCUMENT
C       'PROPOSED PROCEDURE FOR ESTIMATING PRECIPITATING ENERGY
C       FLUX AND AVERAGE ENERGY POLEWARD OF RCM'S MAIN MODELING
C       REGION' DATED 02-27-89.
C
C   PROGRAMMER:  R.W. SPIRO                LAST UPDATE: 11-29-89
C
C   CALLING PARAMETERS:
C       LATDIM      LATITUDINAL DIMENSION
C       LTDIM       LOCAL TIME DIMENSION
C       ISPDIM      MASS SPECIES DIMENSION
C       ITMDIM      TIME LABEL DIMENSION
C       ITMCUR      CURRENT TIME LABEL
C       FKP         FLOATING PT KP VALUE
C       FLXSUM      PRECIPITATING ENERGY FLUX ARRAY IN UNITS OF
C                   ERG/CM**2/S
C       EAVG        AVERAGE ENERGY ARRAY
C       COLAT       COLATITUDE ARRAY (RADIAN)
C       ALOCT       LOCAL TIME ANGLE ARRAY (RADIAN EASTWARD FROM
C                   LOCAL NOON)
C       ALPHA       LATITUDINAL GRID SPACING VECTOR
C       BETA        LOCAL TIME GRID SPACING VECTOR
C       BNDLOC      OUTER BNDY OF DETAILED MSM CALCULATION
C       A,B,DX,DY   PARAMETERS SPECIFYING ELECTRIC FIELD BNDY
C                   ELLIPSES
C
C   PARAMETER(JDIM=51)
C
C   COMMON /GRID/ DLAM,DPSI,RI,JWRAP
C
C   DIMENSION FLXSUM(LATDIM,LTDIM,ISPDIM),EAVG(LATDIM,LTDIM,ISPDIM)
C   DIMENSION COLAT(LATDIM,LTDIM),ALOCT(LATDIM,LTDIM)
C   DIMENSION ALPHA(LATDIM),BETA(LATDIM),BNDLOC(LTDIM,ITMDIM)
C   DIMENSION A(3),B(3),DX(3),DY(3)
```

90/06/30
12:03:44

108

msml.for

```
DIMENSION XMLT(26),PWRKP(26,7),PTOT(JDIM),PWRTOT(JDIM)
DIMENSION FRACT(JDIM),RENORM(JDIM),PWRMDL(JDIM)

C PWRMDL(J)=TOTAL PRECIPITATED POWER WITHIN MODEL REGION PER J GRID LINEMSM68480
C PTOT(J)=OBSERVED POWER PER HOUR OF LOCAL TIME*D(LT)/DJ AT JGRID FOR MSM68490
C      GIVEN KP VALUE MSM68500
C MSM68510
C PWRTOT(J)=OBSERVED POWER CORRECTED FOR OUR MODELING REGION MSM68520
C      =PTOT(J)*(1.-FRACT(J)) MSM68530
C MSM68540
C RENORM(J)=PWRMDL(J)/PWRTOT(J) MSM68550
C MSM68560
C MSM68570
C MSM68580
C MSM68590
C MSM68600
C MSM68610
DATA XMLT/11.5,12.5,13.5,14.5,15.5,16.5,17.5,18.5, MSM68620
$ 19.5,20.5,21.5,22.5,23.5,24.5, MSM68630
$ 25.5,26.5,27.5,28.5,29.5,30.5, MSM68640
$ 31.5,32.5,33.5,34.5,35.5,36.5/ MSM68650
C MSM68660
C PWRKP VALUES COME FROM INTEGRATING HARDY MODEL OVER LATITUDE USING MSM68670
C PROGRAM SUMHDY (NOVEMBER, 1989) MSM68680
C MSM68690
DATA PWRKP /.161,.171,.156,.168,.140,.127,.126,.109,.095,.097, MSM68700
1 .131,.222,.280,.209,.147,.142,.168,.211,.185,.159, MSM68710
1 .173,.189,.174,.156,.161,.171, MSM68720
2 .201,.146,.132,.164,.168,.161,.171,.226,.290,.303, MSM68730
2 .320,.387,.483,.520,.454,.424,.453,.464,.403,.353, MSM68740
2 .346,.326,.285,.243,.201,.146, MSM68750
3 .280,.183,.156,.207,.271,.258,.254,.317,.430,.532, MSM68760
3 .624,.751,.898,1.05,1.14,1.15,.959,.689,.615,.605, MSM68770
3 .564,.515,.455,.388,.280,.183, MSM68780
4 .263,.209,.224,.271,.300,.368,.585,.915,1.09,1.01, MSM68790
4 .973,1.23,1.64,1.83,1.72,1.61,1.63,1.60,1.31,.999, MSM68800
4 .872,.841,.688,.428,.263,.209, MSM68810
5 .263,.189,.166,.195,.215,.360,.699,1.01,1.48,1.65, MSM68820
5 1.57,1.57,1.92,2.18,1.84,1.55,1.76,2.25,2.12,1.49, MSM68830
5 1.21,1.10,.784,.433,.263,.189, MSM68840
6 .287,.229,.265,.340,.550,.880,.969,1.39,2.19,2.24, MSM68850
6 1.72,1.62,2.00,2.41,1.91,1.84,2.58,2.98,2.34,1.69, MSM68860
6 1.32,.978,.644,.425,.287,.229, MSM68870
7 .360,.315,.279,.309,.472,.849,1.60,2.42,3.12,3.27, MSM68880
7 3.15,3.99,6.21,7.26,5.63,4.19,4.45,5.02,4.12,2.89, MSM68890
7 2.06,1.38,.730,.431,.360,.315/ MSM68900
C MSM68910
PI=ATAN2(0.,-1.) MSM68920
DLTDJ=24./FLOAT(LTDIM-JWRAP) MSM68930
BKP=FKP+1. MSM68940
IF(BKP.GT.7.) BKP=7. MSM68950
C MSM68960
C CHECK DIMENSION COMPATIBILITY MSM68970
IF(JDIM.NE.LTDIM) THEN MSM68980
WRITE(6,*) '**** DIMENSION ERROR IN SUBROUTINE PWRCAL ****' MSM68990
WRITE(6,*) ' JDIM=',JDIM,' LTDIM=',LTDIM MSM69000
WRITE(6,*) '**** STOPPING PROGRAM IN PWRCAL ****' MSM69010
STOP MSM69020
END IF MSM69030
C MSM69040
C COMPUTE PTOT AND PWRTOT MSM69050
C MSM69060
C MSM69070
DO 10 J=JWRAP,LTDIM-1 MSM69080
IMIN=NINT(BNDLOC(J,ITMCUR)+.5) MSM69090
FRACT(J)=0.5+.3*SIN(ALOCT(IMIN,J)) MSM69100
TIME=ALOCT(IMIN,J)*12./PI+12. MSM69110
```

90/06/30
12:03:44

msm1.for

109

```
C
DO 20 N=1,25
  IF (TIME.GT.XMLT(N).AND.TIME.LE.XMLT(N+1)) THEN
    F=(TIME-XMLT(N))/(XMLT(N+1)-XMLT(N))
    BXMLT=FLOAT(N)+F
    GO TO 15
  END IF
20 CONTINUE
  WRITE(6,*)'**** ERROR IN PWRCAL ****'
  WRITE(6,*) 'J=',J,' TIME=',TIME
  WRITE(6,*) 'STOPPING PGM IN PWRCAL'
  STOP
15 CONTINUE

C
  PTOT(J)=G3NTRP(PWRKP,26,7,1,BXMLT,BKP,1.)
  PTOT(J)=PTOT(J)*DLTDJ
  PWRTOT(J)=PTOT(J)*(1.-FRACT(J))

C
  PWRMDL(J)=0.

C
  IMAX=LATDIM

C
DO 30 I=IMIN,IMAX
  PWRMDL(J)=PWRMDL(J)+FLXSUM(I,J,1)*ALPHA(I)*BETA(I)
30 CONTINUE
  PWRMDL(J)=1.E-6*DLAM*DPSI*RI**2*PWRMDL(J)
40 CONTINUE

C
C PWRMDL(J) IS IN UNITS OF GW/GRID LINE

  RENORM(J)=PWRMDL(J)/PWRTOT(J)
10 CONTINUE
C CIRCULARIZE
  RENORM(1)=RENORM(JDIM-2)
  RENORM(2)=RENORM(JDIM-1)
  RENORM(JDIM)=RENORM(JWRAP)
  PTOT(1)=PTOT(JDIM-2)
  PTOT(2)=PTOT(JDIM-1)
  PTOT(JDIM)=PTOT(JWRAP)

  FRACT(1)=FRACT(JDIM-2)
  FRACT(2)=FRACT(JDIM-1)
  FRACT(JDIM)=FRACT(JWRAP)

C
  PWRTOT(1)=PWRTOT(JDIM-2)
  PWRTOT(2)=PWRTOT(JDIM-1)
  PWRTOT(JDIM)=PWRTOT(JWRAP)

C
  PWRMDL(1)=PWRMDL(JDIM-2)
  PWRMDL(2)=PWRMDL(JDIM-1)
  PWRMDL(JDIM)=PWRMDL(JWRAP)

C
C DEBUG 6.06.90 $$$$
  CALL OUTP(PWRMDL,1,JDIM,1,1,1,1,JDIM,1,0.0,'PWRMDL',6,132)
  CALL OUTP(PTOT,1,JDIM,1,1,1,1,JDIM,1,0.0,'PTOT',6,132)
  CALL OUTP(PWRTOT,1,JDIM,1,1,1,1,JDIM,1,0.0,'PWRTOT',6,132)
  CALL OUTP(RENORM,1,JDIM,1,1,1,1,JDIM,1,0.0,'RENORM',6,132)
C END DEBUG 6.06.90 $$$$
C
C WRITE(6,*) 'A(1)= ',A(1),' B(1)= ',B(1),' DX(1)= ',DX(1)
C WRITE(6,*) 'DY(1)= ',DY(1)
C
DO 100 J=1,JDIM
```

MSM69120
MSM69130
MSM69140
MSM69150
MSM69160
MSM69170
MSM69180
MSM69190
MSM69200
MSM69210
MSM69220
MSM69230
MSM69240
MSM69250
MSM69260
MSM69270
MSM69280
MSM69290
MSM69300
MSM69310
MSM69320
MSM69330
MSM69340
MSM69350
MSM69360
MSM69370
MSM69380
MSM69390
MSM69400
MSM69410
MSM69420
MSM69430
MSM69440
MSM69450
MSM69460
MSM69470
MSM69480
MSM69490
MSM69500
MSM69510
MSM69520
MSM69530
MSM69540
MSM69550
MSM69560
MSM69570
MSM69580
MSM69590
MSM69600
MSM69610
MSM69620
MSM69630
MSM69640
MSM69650
MSM69660
MSM69670
MSM69680
MSM69690
MSM69700
MSM69710
MSM69720
MSM69730
MSM69740
MSM69750

MSM69760
MSM69770
MSM69780
MSM69790
MSM69800
MSM69810
MSM69820
MSM69830
MSM69840
MSM69850
MSM69860
MSM69870
MSM69880
MSM69890
MSM69900
MSM69910
MSM69920
MSM69930
MSM69940
MSM69950
MSM69960
MSM69970
MSM69980
MSM69990
MSM70000
MSM70010
MSM70020
MSM70030
MSM70040
MSM70050
MSM70060
MSM70070
MSM70080
MSM70090
MSM70100
MSM70110
MSM70120
MSM70130
MSM70140
MSM70150
MSM70160
MSM70170
MSM70180
MSM70190
MSM70200
MSM70210
MSM70220
MSM70230
MSM70240
MSM70250
MSM70260
MSM70270
MSM70280
MSM70290
MSM70300
MSM70310
MSM70320
MSM70330
MSM70340
MSM70350
MSM70360
MSM70370
MSM70380
MSM70390

90/06/30
12:03:44

msm1.for

111

```

      2      +BX*(FIP**3-FIM**3)/3.      MSM70400
      3      +CX*(FIP**4-FIM**4)/4.)      MSM70410
      END IF      MSM70420
C      IF(PS.GE.0.) THEN      MSM70430
      FLXSUM(I,J,1)=1.E6*PS/(ALPHA(I)*BETA(I)*DLAM*DPSI*RI**2)      MSM70440
      EAVG(I,J,1)=(1.7-1.3*COS(ALOCT(I,J)))*1.E3      MSM70450
      ELSE      MSM70460
      GO TO 150      MSM70470
      END IF      MSM70480
C      130 CONTINUE      MSM70490
C      GO TO 200      MSM70500
C      150 CONTINUE      MSM70510
C      TO GET HERE MEANS PS .LT. 0 FOR SOME GRID PT      MSM70520
      ILIM=BIL-.5      MSM70530
      DD=FLOAT(IMIN-ILIM)+.5      MSM70540
      PS=PT/DD      MSM70550
      DO 160 I=ILIM,IMIN-1      MSM70560
      FLXSUM(I,J,1)=1.E6*PS/(ALPHA(I)*BETA(I)*DLAM*DPSI*RI**2)      MSM70570
      EAVG(I,J,1)=(1.7-1.3*COS(ALOCT(I,J)))*1.E3      MSM70580
160  CONTINUE      MSM70590
      FLXSUM(IMIN,J,1)=.5*1.E6*PS/(ALPHA(I)*BETA(I)*DLAM*DPSI*RI**2)      MSM70600
      EAVG(IMIN,J,1)=3.      MSM70610
C      200 CONTINUE      MSM70620
C      DEBUG PRINTOUT PARAMETERS      MSM70630
C      BI1=FLOAT(IMIN+1)-BIL      MSM70640
C      BI2=FLOAT(IMIN+2)-BIL      MSM70650
C      PX1=AX*(BI1+BX*BI1**2+CX*BI1**3)      MSM70660
C      PX2=AX*(BI2+BX*BI2**2+CX*BI2**3)      MSM70670
C      IL=BIL-.5      MSM70680
C      IH=IMIN-1      MSM70690
C      PXT=0.      MSM70700
C      DO 210 I=IL,IH      MSM70710
      PXT=PXT+1.E-6*FLXSUM(I,J,1)*ALPHA(I)*BETA(I)*DLAM*DPSI*RI**2      MSM70720
210  CONTINUE      MSM70730
      BIX=FLOAT(IMIN)-BIL      MSM70740
      BIY=FLOAT(IMIN)-.5-BIL      MSM70750
      PXT=PXT+AX*((BIX**2-BIY**2)/2.+BX*(BIX**3-BIY**3)/3.      MSM70760
      +CX*(BIX**4-BIY**4)/4.)      MSM70770
C      2      MSM70780
C      MSM70790
C      MSM70800
C      100 CONTINUE      MSM70810
C      RETURN      MSM70820
C      END      MSM70830
C      FUNCTION WKRATE(RMID,RPPMID,PMID,ENRG)      MSM70840
C      PURPOSE: FUNCTION SUBPROGRAM TO EVALUATE WEAK PRECIPITATION LOSS      MSM70850
C      RATE AS DESCRIBED IN R.A. WOLF DOCUMENT 'PITCH ANGLE      MSM70860
C      SCATTERING ASSUMPTIONS FOR 1/90 MSM RUNS' DATED 1/4/90.      MSM70870
C      PROGRAMMER: R. W. SPIRO      MSM70880
C      DATE: 01.05.90      MSM70890
C      MSM70900
C      MSM70910
C      MSM70920
C      MSM70930
C      MSM70940
C      MSM70950
C      MSM70960
C      MSM70970
C      MSM70980
C      MSM70990
C      MSM71000
C      MSM71010
C      MSM71020
C      MSM71030
```

90/06/30
12:03:44

msml.for

112

```
C
C INPUT PARAMETERS:
C   RMID          R VALUE AT PT WHERE WKRATE IS TO BE EVALUATED
C                 (RE)
C   RPPMID        ESTIMATE OF PLASMAPAUSE RADIUS (RE)
C   PMJD          LOCAL TIME HOUR ANGLE IN EQUATORIAL PLANE
C                 (RADIAN FROM LOCAL NOON)
C   ENRG          PARTICLE ENERGY (EV)
C
C   IF (RMID.LE.RPPMID) THEN
C USE LYONS RESULTS
      WKRATE=RLYONS (RMID,ENRG)
      GO TO 100
    ELSE IF (ENRG.LE.20.E3) THEN
      WKRATE=0.
      GO TO 100
    ELSE
      WKGEO=MIN (ENRG/20.E3-1.,1.)*(1.+(9./11.)*COS (PMID))/1200.
C
      IF (RMID.GT.6.6) THEN
        WKRATE=WKGEO
        GO TO 100
      ELSE
        XLYONS=RLYONS (RPPMID,ENRG)
        WKRATE=((6.6-RMID)/(6.6-RPPMID))*XLYONS+
2          ((RMID-RPPMID)/(6.6-RPPMID))*WKGEO
      END IF
    END IF
C
100 CONTINUE
    IF (WKRATE.GT.10.) THEN
      WRITE (6,800) RMID,RPPMID,PMID,ENRG,WKRATE
800 FORMAT (1X,'WKRATE.GT.10'/1X,'RMID=',F6.3,2X,'RPPMID=',F6.3,2X,
2       'PMID=',F6.3,2X,'ENRG=',1PE12.3,2X,'WKRATE=',E12.3)
    END IF
    RETURN
    END
    REAL FUNCTION RLYONS (RMID,ENRG)
C
C PURPOSE: FUNCTION SUBPROGRAM TO EVALUATE LYONS-BASED WEAK LOSS RATE.
C
C PROGRAMMER: R.W. SPIRO          DATE: 01.05.90
C
    IF (ENRG.LE.50.E3) THEN
      XLOGLO=(4.9365+15.8621)-6.06897*RMID+0.551724*RMID**2
      XLOGHI=(4.9365+13.5172)-5.58621*RMID+0.551724*RMID**2
      F=(ENRG-20.E3)/(50.E3-20.E3)
    ELSE IF (ENRG.GT.50.E3.AND.ENRG.LE.200.E3) THEN
      XLOGLO=(4.9365+13.5172)-5.58621*RMID+0.551724*RMID**2
      XLOGHI=(4.9365+1.15517)-0.42069*RMID+0.0275862*RMID**2
      F=(ENRG-50.E3)/(200.E3-50.E3)
    ELSE
      XLOGLO=(4.9365+1.15517)-0.42069*RMID+0.0275862*RMID**2
      XLOGHI=(4.9365+3.89655)-1.75862*RMID+0.206897*RMID**2
      F=(ENRG-200.E3)/(500.E3-200.E3)
    END IF
C
    XLOG=(1.-F)*XLOGLO+F*XLOGHI
C
    TAU=10.**XLOG
C
    XLYONS=1./TAU
    RLYONS=XLYONS
C
```

MSM71040
MSM71050
MSM71060
MSM71070
MSM71080
MSM71090
MSM71100
MSM71110
MSM71120
MSM71130
MSM71140
MSM71150
MSM71160
MSM71170
MSM71180
MSM71190
MSM71200
MSM71210
MSM71220
MSM71230
MSM71240
MSM71250
MSM71260
MSM71270
MSM71280
MSM71290
MSM71300
MSM71310
MSM71320
MSM71330
MSM71340
MSM71350
MSM71360
MSM71370
MSM71380
MSM71390
MSM71400
MSM71410
MSM71420
MSM71430
MSM71440
MSM71450
MSM71460
MSM71470
MSM71480
MSM71490
MSM71500
MSM71510
MSM71520
MSM71530
MSM71540
MSM71550
MSM71560
MSM71570
MSM71580
MSM71590
MSM71600
MSM71610
MSM71620
MSM71630
MSM71640
MSM71650
MSM71660
MSM71670

90/06/30
12:03:44

msm1.for

113

```
C      RETURN
C      END
C      FUNCTION G3TRPA(A, IMAX, JMAX, KMAX, BI, BJ, BK)
C
C  VERSION 1.00                                DATE: 01.09.90
C
C  PURPOSE: FUNCTION SUBPROGRAM TO PERFORM A GENERAL 3-D LINEAR
C  INTERPOLATION OF ANGULAR ARRAY A(I,J,K) AT PT(BV(1),BV(2),BV(3))
C
C  INPUT:
C      A                                3-D ARRAY TO BE INTERPOLATED
C                                     INTERPOLATE A
C
C      COMMON /LUNIT/LUERR, LUPIEN, LUCORD, LUPRNT, LUIDAT, LUHDYE,
1      LUDK, LUFLXB, LUEBEG, LUSHFT, LUPIFX, LUEAVG, LUFLSM, LUFLX,
2      LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,
3      LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH
C      DIMENSION A(IMAX, JMAX, KMAX)
C      DIMENSION NDX(3), NDIM(3), BV(3), COEF(3,2)
C
C  PREPARE INDICES FOR INTERPOLATION
C
C      PI=ATAN2(0., -1.)
C      NDIM(1)=IMAX
C      NDIM(2)=JMAX
C      NDIM(3)=KMAX
C      BV(1)=BI
C      BV(2)=BJ
C      BV(3)=BK
C      DO 10 L=1,3
C          NDX(L)=BV(L)
C          IF (NDX(L).LT.1) NDX(L)=1
C          IF (NDX(L).GT.NDIM(L)-1) NDX(L)=NDIM(L)-1
C          IF (NDX(L).LE.0) NDX(L)=1
C
C      FNDX=REAL(NDX(L))
C      COEF(L,1)=1.-BV(L)+FNDX
C      COEF(L,2)=BV(L)-FNDX
10  CONTINUE
C
C      G3TRPA=0.
C      IC=0
C      KSTOP = MIN(KMAX,2)
C      JSTOP = MIN(JMAX,2)
C      DO 20 I=1,2
C          DO 20 J=1, JSTOP
C              DO 20 K=1, KSTOP
C                  IC=IC+1
C                  AVAL=A(NDX(1)+I-1, NDX(2)+J-1, NDX(3)+K-1)
C                  IF (IC.EQ.1) THEN
C  USE FIRST VALUE OF ARRAY TO SET MODULUS
C                      REFMOD=AVAL
C                      END IF
C
C                  IF ((AVAL-REFMOD).GT.PI) THEN
C                      AVAL=AVAL-2.*PI
C                  ELSE IF ((AVAL-REFMOD).LE.-PI) THEN
C                      AVAL=AVAL+2.*PI
C                  END IF
C
C      END IF
```

MSM71680
MSM71690
MSM71700
MSM71710
MSM71720
MSM71730
MSM71740
MSM71750
MSM71760
MSM71770
MSM71780
MSM71790
MSM71800
MSM71810
MSM71820
MSM71830
MSM71840
MSM71850
MSM71860
MSM71870
MSM71880
MSM71890
MSM71900
MSM71910
MSM71920
MSM71930
MSM71940
MSM71950
MSM71960
MSM71970
MSM71980
MSM71990
MSM72000
MSM72010
MSM72020
MSM72030
MSM72040
MSM72050
MSM72060
MSM72070
MSM72080
MSM72090
MSM72100
MSM72110
MSM72120
MSM72130
MSM72140
MSM72150
MSM72160
MSM72170
MSM72180
MSM72190
MSM72200
MSM72210
MSM72220
MSM72230
MSM72240
MSM72250
MSM72260
MSM72270
MSM72280
MSM72290
MSM72300
MSM72310

90/06/30
12:03:44

msm1.for

114

```
C
      G3TRPA=G3TRPA+
1      COEF (1,I)*COEF (2,J)*COEF (3,K)*AVAL
20 CONTINUE
C
C GET RESULT IN A REASONABLE MODULUS
      IF (G3TRPA.GT.2.*PI) THEN
        G3TRPA=G3TRPA-2.*PI
      ELSE IF (G3TRPA.LT.0.) THEN
        G3TRPA=G3TRPA+2.*PI
      END IF
      RETURN
      END
      SUBROUTINE THRCAL (FKP,FNRGLG,NRGNDX,KPNDX,IRDIM,NRGDIM,KPDIM,
2      KPPLUS,ISPDIM,FLXMAT,THR)
C
C PURPOSE: SUBROUTINE TO CALCULATE THRESHOLD FLUX AT R=3,4,6.6, AND
C          13 RE FOR KP=FKP AND LOG(ENERGY)=FNRGLG. THR IS IN UNITS
C          OF PARTICLES/CM**2/S/SR/EV.
C
C PROGRAMMER: R.W. SPIRO          DATE: 05.09.90
C
      DIMENSION FLXMAT (IRDIM,NRGDIM,KPPLUS,ISPDIM), THR (4),RTHR (4)
      DIMENSION EZERO (4,7), RKAP (4)
C
      DATA RTHR /3.,4.,6.6,13./
C
      DATA EZERO /23.5E3,16.E3,21.E3,.49275E3,
1      23.5E3,16.E3,4.8324E3,.625E3,
2      23.5E3,16.E3,4.814E3,.75725E3,
3      23.5E3,16.E3,4.7978E3,.8895E3,
4      23.5E3,16.E3,4.816E3,1.02175,
5      23.5E3,16.E3,4.8943E3,1.154E3,
6      23.5E3,16.E3,5.0752E3,1.28625E3/
C
      DATA RKAP /2.8,2.0,6.,6./
C
      ETOP=40.E3
      ELOW=20.E3
      ENRG=10.**FNRGLG
C
      IF (ENRG.LE.ELOW) THEN
        THR (1)=FLXMAT (1,NRGNDX,1,1)
        THR (2)=FLXMAT (2,NRGNDX,1,1)
        THR (3)=FLXMAT (3,NRGNDX,1,1)
        THR (4)=FLXMAT (4,NRGNDX,1,1)
C
      ELSE IF (ENRG.GE.ETOP) THEN
        DO 10 N=1,4
          RK=RKAP (N)
          EZ=EZERO (N,KPNDX)
C
          IF (RK.GE.6.0) THEN
            ER=(1./EZ)*((RK-1.)/RK)*(ENRG/EZ)*(1.+ETOP/(RK*EZ))*RK/
2            (1.+MIN (ENRG/RK/EZ,1.E5))*((RK+1.)/(1.+ETOP/EZ))
          ELSE
            ER=(1./EZ)*((RK-1.)/RK)*(ENRG/EZ)*(1.+ETOP/(RK*EZ))*RK/
2            (1.+ENRG/RK/EZ)*((RK+1.)/(1.+ETOP/EZ))
          ENDDIF
C
          THR (N)=ALOG10 (5.E7*ER*(6.6/RTHR (N))**4)
C
```

MSM72320
MSM72330
MSM72340
MSM72350
MSM72360
MSM72370
MSM72380
MSM72390
MSM72400
MSM72410
MSM72420
MSM72430
MSM72440
MSM72450
MSM72460
MSM72470
MSM72480
MSM72490
MSM72500
MSM72510
MSM72520
MSM72530
MSM72540
MSM72550
MSM72560
MSM72570
MSM72580
MSM72590
MSM72600
MSM72610
MSM72620
MSM72630
MSM72640
MSM72650
MSM72660
MSM72670
MSM72680
MSM72690
MSM72700
MSM72710
MSM72720
MSM72730
MSM72740
MSM72750
MSM72760
MSM72770
MSM72780
MSM72790
MSM72800
MSM72810
MSM72820
MSM72830
MSM72840
MSM72850
MSM72860
MSM72870
MSM72880
MSM72890
MSM72900
MSM72910
MSM72920
MSM72930
MSM72940
MSM72950

90/06/30
12:03:44

msm1.for

115

```
10    CONTINUE
C
      ELSE
C    ENRG IS BETWEEN ELOW AND ETOP
C
      DO 20 N=1, 4
        RK=RKAP (N)
        EZ=EZERO (N, KPNDX)
C
        ERTOP=(1./EZ)*((RK-1.)/RK)*(ETOP/EZ)/
          (1.+ETOP/RK/EZ)/(1.+ETOP/EZ)
C
        THRTOP=ALOG10(5.E7*ERTOP*(6.6/RTHR(N))**4)
C
        THRLOW=FLXMAT (N, NRGNDX, 1, 1)
C
        THR(N) = (ETOP-ENRG)*THRLOW+(ENRG-ELOW)*THRTOP/(ETOP-ELOW)
C
20    CONTINUE
C
      END IF
C
      RETURN
      END
      SUBROUTINE FLXDFL (ITM, FKP, R, P, ALAM, VM, IRDIM, NRGDIM, KPDIM, KPPLUS,
1    ISPDIM, FLXMAT, FLXR, FLXNRG, FLXKP, EFLUX, ETA, ALMDEL, LATDIM,
2    LTDIM, IEDIM, ITMDIM, BNDLOC, IEMAX, IFLAV, IELEG, IEEND)
C
C    SUBROUTINE WHICH USES THE FLXMAT ARRAYS TO
C    CALCULATE THE ENERGY-DEPENDENT ETA ARRAYS.  THIS IS THE
C    DEFAULT MODEL WHEN ONLY KP IS AVAILABLE TO RUN THE PROGRAM.
C
      PARAMETER (JJDIM=51)
      DIMENSION R (LATDIM, LTDIM, ITMDIM), P (LATDIM, LTDIM, ITMDIM),
1    ALAM (IEDIM), VM (LATDIM, LTDIM, ITMDIM),
2    FLXMAT (IRDIM, NRGDIM, KPPLUS, ISPDIM),
3    FLXR (IRDIM), FLXNRG (NRGDIM), FLXKP (KPDIM),
4    BNDLOC (LTDIM, ITMDIM), EFLUX (LATDIM, LTDIM, IEDIM), IBLOC (JJDIM),
5    ETA (LATDIM, LTDIM, IEDIM), ALMDEL (IEDIM), IFLAV (IEDIM)
C
      IF (JJDIM.NE.LTDIM) THEN
        WRITE(6,*)' DIMENSION ERROR IN FLXDFL SUBROUTINE, PROGRAM STOPS.'
        STOP
      ENDIF
C
      DO 10 J=1, LTDIM
        IBLOC (J)=IFIX (BNDLOC (J, ITM))
        DO 20 I=IBLOC (J), LATDIM
          DO 30 K=IELEG, IEEND
            EFLUX (I, J, K)=0.0
30      CONTINUE
20      CONTINUE
10      CONTINUE
        DO 50 J=1, LTDIM
          DO 60 I=IBLOC (J), LATDIM
            DO 70 K=IELEG, IEEND
              ISP=IFLAV (K)
              ENRG=ABS (ALAM (K)) *VM (I, J, ITM)
              FLXOUT=FLXVAL (ISP, FKP, R (I, J, ITM), P (I, J, ITM), ENRG, IRDIM,
1            NRGDIM, KPDIM, KPPLUS, ISPDIM, FLXMAT, FLXR, FLXNRG, FLXKP)
              IF (IFLAV (K).EQ.1) THEN
                XMSCNT=7.392E-16
C
                IF ((K.EQ.1).OR.(K.EQ.5).OR.(K.EQ.9).OR.(K.EQ.13)) THEN
C
                  WRITE(6,*)' I, J, K, FLUX, ENRG ', I, J, K, FLXOUT, ENRG

```

MSM72960
MSM72970
MSM72980
MSM72990
MSM73000
MSM73010
MSM73020
MSM73030
MSM73040
MSM73050
MSM73060
MSM73070
MSM73080
MSM73090
MSM73100
MSM73110
MSM73120
MSM73130
MSM73140
MSM73150
MSM73160
MSM73170
MSM73180
MSM73190
MSM73200
MSM73210
MSM73220
MSM73230
MSM73240
MSM73250
MSM73260
MSM73270
MSM73280
MSM73290
MSM73300
MSM73310
MSM73320
MSM73330
MSM73340
MSM73350
MSM73360
MSM73370
MSM73380
MSM73390
MSM73400
MSM73410
MSM73420
MSM73430
MSM73440
MSM73450
MSM73460
MSM73470
MSM73480
MSM73490
MSM73500
MSM73510
MSM73520
MSM73530
MSM73540
MSM73550
MSM73560
MSM73570
MSM73580
MSM73590

90/06/30
12:03:44

msml.for

116

```
C      WRITE(6,*)' VM,ALMDEL,MSCNST',VM(I,J,ITM),ALMDEL(K),MSCNST
C      WRITE(6,*)' IFLAV ',IFLAV(K)
C      ENDIF
C      ETA(I,J,K)=(FLXOUT*ALMDEL(K))/(XMSCNT*SQRT(ABS(ALAM(K)))
1          *VM(I,J,ITM))
C      IF((K.EQ.1).OR.(K.EQ.5).OR.(K.EQ.9).OR.(K.EQ.13))THEN
C      WRITE(6,*)' ETA ',ETA(I,J,K)
C      ENDIF
C      ELSE
C      IONS
C      XMSCNT=1.731E-17
C      IF(IFLAV(K).EQ.2) THEN
C      ATOMWT=1.0
C      ELSE
C      ATOMWT=16.0
C      ENDIF
C      ETA(I,J,K)=(FLXOUT*ALMDEL(K)*SQRT(ATOMWT))/(XMSCNT*
1          SQRT(ABS(ALAM(K)))*VM(I,J,ITM))
C      ENDIF
70      CONTINUE
60      CONTINUE
50      CONTINUE
C      DO 100 K=IEBEG,IEEND
C      IF((K.EQ.1).OR.(K.EQ.5).OR.(K.EQ.9).OR.(K.EQ.13))THEN
C      WRITE(18,*)' K = ',K
C      WRITE(6,*)' K = ',K
C      CALL OUTP(ETA(1,1,K),LATDIM,LTDIM,10,50,1,3,LTDIM-1,1,
C      1      0.,'ETA',6,132)
C      ENDIF
100     CONTINUE
        RETURN
        END
        SUBROUTINE RMVBSH(WORK,LATDIM,LTDIM)
C
C      TEMPORARY FIX TO THE B-FIELD MATRICES TO REMOVE THE LARGE DATA
C      VALUES SIGNIFYING THAT THE FIELD LINES HAD 'GONE INTO THE BUSHES'.
C      IT LOOKS FOR THE CLOSEST VALUE IN J AND LINEARLY INTERPOLATES FOR
C      THE I VALUES.
C
        DIMENSION WORK(LATDIM,LTDIM,2,2,2,2)
        BFBUSH = 1.E20
        DO 10 N1=1,2
        DO 20 N2=1,2
        DO 30 N3=1,2
        DO 40 N4=1,2
        DO 50 N5=1,2
        DO 60 J=1,LTDIM
        DO 70 I=1,LATDIM
        IF(WORK(I,J,N1,N2,N3,N4,N5).LT.BFBUSH)GO TO 70
        IDO=I
999      CONTINUE
        IDO=IDO+1
        IF(WORK(IDO,J,N1,N2,N3,N4,N5).GT.BFBUSH)GO TO 999
        COEF1=WORK(IDO,J,N1,N2,N3,N4,N5)
        COEF2= WORK(IDO+1,J,N1,N2,N3,N4,N5)
        STRTLN=COEF2-COEF1
        DO 80 ICHK=IDO-1,1,-1
        WORK(ICHK,J,N1,N2,N3,N4,N5)=WORK(ICHK+1,J,N1,N2,N3,N4,N5)
1          - STRTLN
80      CONTINUE
70      CONTINUE
60      CONTINUE
50      CONTINUE
40      CONTINUE
```

MSM73600
MSM73610
MSM73620
MSM73630
MSM73640
MSM73650
MSM73660
MSM73670
MSM73680
MSM73690
MSM73700
MSM73710
MSM73720
MSM73730
MSM73740
MSM73750
MSM73760
MSM73770
MSM73780
MSM73790
MSM73800
MSM73810
MSM73820
MSM73830
MSM73840
MSM73850
MSM73860
MSM73870
MSM73880
MSM73890
MSM73900
MSM73910
MSM73920
MSM73930
MSM73940
MSM73950
MSM73960
MSM73970
MSM73980
MSM73990
MSM74000
MSM74010
MSM74020
MSM74030
MSM74040
MSM74050
MSM74060
MSM74070
MSM74080
MSM74090
MSM74100
MSM74110
MSM74120
MSM74130
MSM74140
MSM74150
MSM74160
MSM74170
MSM74180
MSM74190
MSM74200
MSM74210
MSM74220
MSM74230

90/06/30
12:03:44

msml.for

117

```
30      CONTINUE
20      CONTINUE
10      CONTINUE
      RETURN
      END
      SUBROUTINE DSTDFL (MODE, AUGPAR, NAUGEL, ITMDIM, ITM, ITMMAX, TIMTAG)
C
C      SUBROUTINE TO CALCULATE THE DEFAULT DST VALUES IF NO
C      VALUE IS FOUND IN THE ENVIRONMENTAL DATA BASE. (MODE IS FALSE)
C
      INTEGER IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST, DDST,
1      EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2      GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3      STAND, PCP, DEQDT, IPATT
C
      COMMON /ARINDX/ IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST,
1      EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2      GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3      STAND, PCP, IPATT, DDST, DEQDT
C
      DIMENSION AUGPAR (NAUGEL, ITMDIM), TIMTAG (ITMDIM)
      DIMENSION YA (2), XA (2)
C
      LOGICAL*1 MODE
      IF (MODE) RETURN
      IF (AUGPAR (KP, ITM) .LT. 4.0) THEN
        AUGPAR (DST, ITM) = -AUGPAR (KP, ITM) * 4.0
      ELSE
        AUGPAR (DST, ITM) = -20.0 - (((AUGPAR (KP, ITM) * 10.0) - 40.0) * 2.6)
      ENDIF
      IF (ITM.GT.1) THEN
        YA (1) = AUGPAR (DST, ITM-1)
        YA (2) = AUGPAR (DST, ITM)
        XA (1) = TIMTAG (ITM-1)
        XA (2) = TIMTAG (ITM)
        CALL DTNTRP (TIMTAG (ITM), 2, XA, YA, Y, DYDX, DELTA, IERR)
        AUGPAR (DDST, ITM-1) = DYDX * 3600.
      ENDIF
      IF (ITM.EQ.ITMMAX) AUGPAR (DDST, ITM) = AUGPAR (DDST, ITM-1)
      RETURN
      END
      SUBROUTINE EQTDFL (MODE, AUGPAR, NAUGEL, ITMDIM, ITM, ITMMAX, TIMTAG)
C
C      SUBROUTINE TO CALCULATE A KP-BASED VALUE FOR THE
C      EQUATORWARD EDGE OF THE AURORAL OVAL BASED ON THE WORK OF
C      GUSSENHOVEN ET. AL., 1983.
C      IF MODE IS TRUE A VALUE HAS ALREADY BEEN RECEIVED FROM THE
C      ENVIRONMENTAL DATA BASE AND THE SUBROUTINE JUST RETURNS.
C
      INTEGER IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST, DDST,
1      EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2      GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3      STAND, PCP, DEQDT, IPATT
C
      COMMON /ARINDX/ IYEAR, IDAY, ISECND, IPHOUR, IPMIN, IPSEC, KP, DST,
1      EQEDGE, DLATAZ, DLATRV, MLTRV, IMFBX, IMFBY, IMFBZ, CLAPSE,
2      GEOMGX, GEOMGY, GEOMGZ, GEOMGT, LBOUND, SWVEL, SWDEN, TILTW,
3      STAND, PCP, IPATT, DDST, DEQDT
C
      DIMENSION AUGPAR (NAUGEL, ITMDIM), TIMTAG (ITMDIM)
      DIMENSION XA (2), YA (2)
C
      LOGICAL MODE*1
      IF (MODE) RETURN
```

MSM74240
MSM74250
MSM74260
MSM74270
MSM74280
MSM74290
MSM74300
MSM74310
MSM74320
MSM74330
MSM74340
MSM74350
MSM74360
MSM74370
MSM74380
MSM74390
MSM74400
MSM74410
MSM74420
MSM74430
MSM74440
MSM74450
MSM74460
MSM74470
MSM74480
MSM74490
MSM74500
MSM74510
MSM74520
MSM74530
MSM74540
MSM74550
MSM74560
MSM74570
MSM74580
MSM74590
MSM74600
MSM74610
MSM74620
MSM74630
MSM74640
MSM74650
MSM74660
MSM74670
MSM74680
MSM74690
MSM74700
MSM74710
MSM74720
MSM74730
MSM74740
MSM74750
MSM74760
MSM74770
MSM74780
MSM74790
MSM74800
MSM74810
MSM74820
MSM74830
MSM74840
MSM74850
MSM74860
MSM74870

90/06/30
12:03:44

msml.for

118

```
AUGPAR(EQEDGE,ITM) = 67.78 - (2.068 * AUGPAR(KP,ITM))
IF (ITM.GT.1) THEN
  YA(1)=AUGPAR(EQEDGE,ITM-1)
  YA(2)=AUGPAR(EQEDGE,ITM)
  XA(1)=TIMTAG(ITM-1)
  XA(2)=TIMTAG(ITM)
  CALL DTNTRP(TIMTAG(ITM),2,XA,YA,Y,DYDX,DELTA,IERR)
  AUGPAR(DEQDT,ITM-1)=DYDX*3600.
ENDIF
IF (ITM.EQ.ITMMAX) AUGPAR(DEQDT,ITM)=AUGPAR(DEQDT,ITM-1)
RETURN
END
SUBROUTINE PCPDFL(MODE,XKP,PCP)
C
C THIS SUBROUTINE CALCULATES THE CROSS-POLAR CAP POTENTIAL (IN KV)
C BASED ON THE WORK OF REIFF ET AL IF MODE IS FALSE. IF MODE IS TRUE
C THE POTENTIAL HAS ALREADY BEEN RECEIVED FROM THE ENVIRONMENTAL DATA
C BASE AND THE SUBROUTINE JUST RETURNS.
C
  LOGICAL*1 MODE
  IF (MODE) RETURN
  PCP = 14.587 + (1.7 * (XKP * 10.0))
  RETURN
  END
  SUBROUTINE PATDFL(MODE,MODEBY,MODEBZ,BY,BZ,IPATT)
C
C THIS SUBROUTINE USES THE BY AND BZ COMPONENTS OF THE IMF TO
C DETERMINE THE ELECTRIC FIELD PATTERN TYPE TO BE USED.
C THE DIFFERENT TYPES ARE:
C BZ>0 - TYPE 0.0
C BZ<0,BY>0 - TYPE 1.0
C BZ<0,BY=0 - TYPE 2.0 (DEFAULT)
C BZ<0,BY<0 - TYPE 3.0
C IF MODE IS TRUE, THE PATTERN TYPE HAS ALREADY BEEN RECEIVED FROM
C THE ENVIRONMENTAL DATA BASE AND THE SUBROUTINE JUST RETURNS.
C
  LOGICAL*1 MODE, MODEBY,MODEBZ
  IF (MODE) RETURN
  IF (.NOT. (MODEBZ)) THEN
    XIPATT=2.0
  ELSE
    IF (BZ.GT.0.0) THEN
      XIPATT=0.0
    ELSE
      IF (.NOT. (MODEBY)) THEN
        XIPATT=2.0
      ELSE
        IF (BY.GT.0.0) XIPATT=1.0
        IF (BY.EQ.0.0) XIPATT=2.0
        IF (BY.LT.0.0) XIPATT=3.0
      ENDIF
    ENDIF
  ENDIF
  RETURN
  END
  SUBROUTINE CLPDFL(MODE,CLAPSE)
C
C CLPASE IS THE INDICATOR FOR WHETHER OR NOT TO USE THE COLLAPSED
C TAIL VERSION OF THE MAGNETIC FIELD MODEL. IF MODE IS FALSE, NO
C VALUE HAS BEEN RETURNED FROM THE ENVIRONMENTAL DATA BASE. THE
C DEFAULT VALUE IS NO COLLAPSE (CLAPSE=1.0).
C
  LOGICAL*1 MODE
  IF (MODE) RETURN
```

MSM74880
MSM74890
MSM74900
MSM74910
MSM74920
MSM74930
MSM74940
MSM74950
MSM74960
MSM74970
MSM74980
MSM74990
MSM75000
MSM75010
MSM75020
MSM75030
MSM75040
MSM75050
MSM75060
MSM75070
MSM75080
MSM75090
MSM75100
MSM75110
MSM75120
MSM75130
MSM75140
MSM75150
MSM75160
MSM75170
MSM75180
MSM75190
MSM75200
MSM75210
MSM75220
MSM75230
MSM75240
MSM75250
MSM75260
MSM75270
MSM75280
MSM75290
MSM75300
MSM75310
MSM75320
MSM75330
MSM75340
MSM75350
MSM75360
MSM75370
MSM75380
MSM75390
MSM75400
MSM75410
MSM75420
MSM75430
MSM75440
MSM75450
MSM75460
MSM75470
MSM75480
MSM75490
MSM75500
MSM75510

90/06/30
12:03:44

msm1.for

119

```
CLAPSE=1.0
RETURN
END
SUBROUTINE STNDFL(MODE,MODVEL,MODDEN,SWVEL,SWDEN,XKP,STAND)
C
C THIS SUBROUTINE CALCULATES THE STANDOFF DISTANCE. IF SOLAR WIND
C VELOCITY AND DENSITY DATA IS AVAILABLE SUBROUTINE STNDOF DOES THE
C THE ACTUAL CALCULATION. OTHERWISE THE DEFAULT IS A KP-BASED
C FORMULA.
C
  LOGICAL*1 MODE,MODVEL,MODDEN
  IF (MODE) RETURN
  IF (MODVEL.AND.MODDEN) THEN
    CALL STNDOF(SWVEL,SWDEN,STAND)
  ELSE
    IF (XKP.LT.6.7) STAND=11.7-XKP
    IF (XKP.GE.6.7) STAND=5.0
  ENDIF
  RETURN
END
SUBROUTINE TILT(DAY,YEAR,XTILT)
C
C SUBROUTINE TO CALCULATE THE EARTH'S TILT ANGLE. CURRENTLY SET
C TO 0.0
C
  XTILT=0.0
  RETURN
END
-----
C
C      =====
C      SUBROUTINE AURL2S (FKP,COLAT,ALOCT,LATDIM,LTDIM,
C      *                AEFLUX,AEMEAN)
C      =====
C      [Inputs]
C      FKP : The most recent Kp value
C      COLAT:An array listing pertinent colatitudes <40 deg
C      ALOCT:An array listing pertinent magnetic loca time
C      [Outputs]
C      AEFLUX:The integral ION energy flux (erg/cm2/sec)
C      AEMEAN:AVERAGE ENERGY IN EV
C      AFLUX :The integral ION number flux (/cm2/sec)
C      This subroutine calls REGEN() and EFUN()
C      June 04, 1989   by Akira NAGAI
C
C      Note: The feature is optimized for the speed.
C            At least 10KB static memory area is required
C            for REAL*2 in MSM Model.
C
C      PARAMETER (IIDIM=62,JJDIM=51)
C
C      COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
1      LUDK,LUFLXB,LUEBEG,LUSHFT,LUIFX,LUEAVG,LUFLSM,LUFLX,
2      LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
3      LUCOLT,LUBNDL,LUPDAT,LUVN,LUVS,LIONEG,LIONNO,LUENCH
C
C      DIMENSION COLAT(LATDIM,LTDIM),ALOCT(LATDIM,LTDIM),
1      AFLUX(IIDIM,JJDIM),AEFLUX(LATDIM,LTDIM),
2      AEMEAN(LATDIM,LTDIM),
3      X1(JJDIM,0:6,2),X2(JJDIM,0:6,2),X3(JJDIM,0:6,2),
4      X4(JJDIM,0:6,2),X5(JJDIM,0:6,2),X6(JJDIM,0:6,2),
5      C1(0:12),C2(0:12),C3(0:12),C4(0:12),C5(0:12),C6(0:12)
```

MSM75520
MSM75530
MSM75540
MSM75550
MSM75560
MSM75570
MSM75580
MSM75590
MSM75600
MSM75610
MSM75620
MSM75630
MSM75640
MSM75650
MSM75660
MSM75670
MSM75680
MSM75690
MSM75700
MSM75710
MSM75720
MSM75730
MSM75740
MSM75750
MSM75760
MSM75770
MSM75780
MSM75790
MSM75800
MSM75810
MSM75820
MSM75830
MSM75840
MSM75850
MSM75860
MSM75870
MSM75880
MSM75890
MSM75900
MSM75910
MSM75920
MSM75930
MSM75940
MSM75950
MSM75960
MSM75970
MSM75980
MSM75990
MSM76000
MSM76010
MSM76020
MSM76030
MSM76040
MSM76050
MSM76060
MSM76070
MSM76080
MSM76090
MSM76100
MSM76110
MSM76120
MSM76130
MSM76140
MSM76150

90/06/30
12:03:44

msml.for

120

```

      IKP=NINT(FKP)
      IF((IPASS.EQ.0).AND.(IKP.EQ.0)) GOTO 60
      IF(IKP.EQ.KP) GOTO 80
60    KP=IKP
      IF(KP.GT.6) KP=6

C      Just once is OK for this loop(10) for the entire run.
      DO 10 ICASE=1,2

          IF(IPASS.EQ.1) GOTO 80

          IF(ICASE.EQ.1) THEN
              OPEN(UNIT=LIONNO,ACCESS='SEQUENTIAL',FILE='IONNUM',
1              STATUS='OLD')
          ELSE
              OPEN(UNIT=LIONEG,ACCESS='SEQUENTIAL',FILE='IONENG',
1              STATUS='OLD')
          ENDIF

C      DO 20 IKP=0,6
          DO 30 J=0,12
              IF(ICASE.EQ.1) THEN
                  READ(LIONNO,*) IDX,C1(J),C2(J),C3(J),C4(J),C5(J),C6(J)
              ELSE
                  READ(LIONEG,*) IDX,C1(J),C2(J),C3(J),C4(J),C5(J),C6(J)
              ENDIF
30      CONTINUE
          CALL REGEN(LATDIM,LTDIM,ALOCT,C1,X1,JJDIM,IKP,ICASE)
          CALL REGEN(LATDIM,LTDIM,ALOCT,C2,X2,JJDIM,IKP,ICASE)
          CALL REGEN(LATDIM,LTDIM,ALOCT,C3,X3,JJDIM,IKP,ICASE)
          CALL REGEN(LATDIM,LTDIM,ALOCT,C4,X4,JJDIM,IKP,ICASE)
          CALL REGEN(LATDIM,LTDIM,ALOCT,C5,X5,JJDIM,IKP,ICASE)
          CALL REGEN(LATDIM,LTDIM,ALOCT,C6,X6,JJDIM,IKP,ICASE)
20      CONTINUE
          CLOSE(LIONNO)
          CLOSE(LIONEG)
10      CONTINUE

      IPASS=1

80      DO 40 IMLT=1,LTDIM
          DO 50 ILAT=1,LATDIM
              H=90.-ABS(COLAT(ILAT,IMLT))*180/3.14159
              FX=EFUN(1,KP,H,X1(IMLT,KP,1),X2(IMLT,KP,1),
*              X3(IMLT,KP,1),X4(IMLT,KP,1),X5(IMLT,KP,1),X6(IMLT,KP,1))
C              Unit from 1/cm2/sec/str to 1/cm2/sec
              AFLUX(ILAT,IMLT) = (10.**FX)*3.1415265358
              FX=EFUN(2,KP,H,X1(IMLT,KP,2),X2(IMLT,KP,2),
*              X3(IMLT,KP,2),X4(IMLT,KP,2),X5(IMLT,KP,2),X6(IMLT,KP,2))
C              Unit from keV/cm2/sec/str to erg/cm2/sec
              AEFLUX(ILAT,IMLT) = (10.**FX)*3.14159265358*1.6021E-9
              IF(AFLUX(ILAT,IMLT).NE.0.0) AEMEAN(ILAT,IMLT) =
1              AEFLUX(ILAT,IMLT)/AFLUX(ILAT,IMLT)
              AEMEAN(ILAT,IMLT)=AEMEAN(ILAT,IMLT)*6.24E11
50      CONTINUE
40      CONTINUE

99      RETURN
      END

C-----
C      *****
C      FUNCTION EFUN(ICASE,KP,H,R0,H0,H1,S0,R1,S2)
C      *****
C      This function is called only by AURL2S() and this does not
```

MSM76160
MSM76170
MSM76180
MSM76190
MSM76200
MSM76210
MSM76220
MSM76230
MSM76240
MSM76250
MSM76260
MSM76270
MSM76275
MSM76280
MSM76290
MSM76295
MSM76300
MSM76310
MSM76320
MSM76330
MSM76340
MSM76350
MSM76360
MSM76370
MSM76380
MSM76390
MSM76400
MSM76410
MSM76420
MSM76430
MSM76440
MSM76450
MSM76460
MSM76470
MSM76480
MSM76490
MSM76500
MSM76510
MSM76520
MSM76530
MSM76540
MSM76550
MSM76560
MSM76570
MSM76580
MSM76590
MSM76600
MSM76610
MSM76620
MSM76630
MSM76640
MSM76650
MSM76660
MSM76670
MSM76680
MSM76690
MSM76700
MSM76710
MSM76720
MSM76730
MSM76740
MSM76750
MSM76760
MSM76770

90/06/30
12:03:44

msml.for

124

```

      XLO=(ENERGY-HAFCHN(NCHNL))/(HAFCHN(NCHNL-1)-
      HAFCHN(NCHNL))
2
C
48  CONTINUE
      XL1=1.-XLO
      XSTFLX=XLO*SATFLX(KK)+XL1*SATFLX(KK+1)
C
      DO 50 INO=1,LATDIM
      DO 60 JNO=1,LTDIM
        FLXNTR(INO,JNO,1)=FLUX(INO,JNO,KENERGY)
        FLXNTR(INO,JNO,2)=FLUX1(INO,JNO,KENERGY)
60    CONTINUE
50    CONTINUE
      FLUXT=G3NTRP(FLXNTR,LATDIM,LTDIM,2,BI,BJ,TMTRP1)
C
C
      FLUXT=FLUXT*1000.0
      IF(FLUXT.LE.0.0) THEN
        WRITE(6,*)' BAD FLUX VALUE CALCULATED IN SUBROUTINE OPTERR'
        WRITE(6,*)' UNABLE TO CONTINUE'
        WRITE(6,*)' VM, FLUX = ',CALVM,FLUXT
        STOP
      ENDIF
C
      WRITE(6,*) 'K=',KENERGY,' SAT FLUX=',XSTFLX,' MODEL FLUX=',
2      LOG10(FLUXT),' ALAM=',ALAM(KENERGY)
C
      IF(STNDEV(KENERGY).EQ.-1.0) THEN
        STNDEV(KENERGY)=(LOG10(FLUXT)-XSTFLX)**2
      ELSE
        STNDEV(KENERGY)=STNDEV(KENERGY)
1      +((LOG10(FLUXT)-XSTFLX)**2)
      ENDIF
C
      NPTS(KENERGY)=NPTS(KENERGY)+1
      COEF(KENERGY)=COEF(KENERGY)+(LOG10(FLUXT)-XSTFLX)
C      WRITE(6,*)' NOCHNL,STNDEV,COEF,KENERGY ',NOCHNL-1,
C      STNDEV(KENERGY),COEF(KENERGY),KENERGY,FLUXT,XSTFLX
1      ENDIF
45  CONTINUE
      XCHNCK=XCHNL
      IF(ISP.EQ.1) THEN
        READ(99,*,END=400)XCHNL,YEAR,DECDAY,LAT,LONG,ALT,MLT
      ELSE
        READ(98,*,END=400) XCHNL,YEAR,DECDAY,LAT,LONG,ALT,MLT
      END IF
C
      IF(XCHNCK.NE.XCHNL) THEN
        WRITE(6,*)' NUMBER OF SATELLITE ENERGY CHANNELS CHANGED IN'
        WRITE(6,*)' MID-RUN. UNABLE TO COMPUTE ERROR. RETURNING TO'
        WRITE(6,*)' MAIN PROGRAM.'
        RETURN
      ENDIF
      TIMCHK=TCONV2(YEAR,DECDAY,YRCHCK,DYCHCK)
      WRITE(6,*)' NPTS,TIMCHK,TIMTAG,ITM ',NPTS,TIMCHK,TIMTAG(ITM+1),
1      ITM
C
      IF(TIMCHK.LT.TIMTAG(ITM+1))GO TO 200
C
C  COMPUTE STANDARD DEVIATION AND TIME SHIFT FOR THIS TIME INTERVAL
C
400 CONTINUE
C
      IF(ISP.EQ.NSP) THEN

```

MSM78700
MSM78710
MSM78720
MSM78730
MSM78740
MSM78750
MSM78760
MSM78770
MSM78780
MSM78790
MSM78800
MSM78810
MSM78820
MSM78830
MSM78840
MSM78850
MSM78860
MSM78870
MSM78880
MSM78890
MSM78900
MSM78910
MSM78920
MSM78930
MSM78940
MSM78950
MSM78960
MSM78970
MSM78980
MSM78990
MSM79000
MSM79010
MSM79020
MSM79030
MSM79040
MSM79050
MSM79060
MSM79070
MSM79080
MSM79090
MSM79100
MSM79110
MSM79120
MSM79130
MSM79140
MSM79150
MSM79160
MSM79170
MSM79180
MSM79190
MSM79200
MSM79210
MSM79220
MSM79230
MSM79240
MSM79250
MSM79260
MSM79270
MSM79280
MSM79290
MSM79300
MSM79310
MSM79320
MSM79330

90/06/30
12:03:44

msm1.for

125

```
DO 70 KENERGY=1,IEMAX
ENERGY=(ABS (ALAM(KENERGY))*VMGEO)/1000.00
WRITE(6,*)' KENERGY,STNDEV ',KENERGY,STNDEV(KENERGY)
IF(NPTS(KENERGY).NE.0)THEN
  STNDEV(KENERGY)=SQRT(STNDEV(KENERGY)/NPTS(KENERGY))
  WRITE(6,*)' STANDARD DEVIATION (LOG10) = ',STNDEV(KENERGY),
1    ' FOR ENERGY ',ENERGY,' (KEV)'
  COEF(KENERGY)=COEF(KENERGY)/NPTS(KENERGY)
  ERSHFT(KENERGY)=COEF(KENERGY)
  WRITE(6,*)' PARALLEL SHIFT (LOG10) = ',COEF(KENERGY),
1    ' FOR ENERGY ',ENERGY,' (KEV)'
  ELSE
    ERSHFT(KENERGY)=0.
  ENDIF
70 CONTINUE
C
C WRITE STANDARD DEVIATION AND SHIFT TO OUTPUT DATA SET
  IRDOUT=IREC+1
  WRITE(6,*) 'WRITING INTO SHFT FILE  IRDOUT=',IRDOUT,
2    ' IEMAX=',IEMAX,' ERSHFT=',ERSHFT
  CALL WRTVEC(LUSHFT,IRDOUT,IEMAX,IEDIM,ALAM,THRSH,ERSHFT,IFLAV)
  END IF
C
30 CONTINUE
C
15 CONTINUE
C
  RETURN
  END
  SUBROUTINE WRTVEC(NOUNIT,IRDOUT,IEMAX,IEDIM,ALAM,THRSH,
1  ERSHFT,IFLAV)
C
C SUBROUTINE TO WRITE
C  ALAMS,THRESHOLD ENERGIES, AND ERROR
C  FOR EACH OF THE ENERGY CHANNELS IN THIS RUN
C
C
C  DIMENSION ALAM(IEDIM),THRSH(IEDIM),ERSHFT(IEDIM),IFLAV(IEDIM)
C
C  IRECL=4*IEDIM+1
C  OPEN(UNIT=NOUNIT,ACCESS='DIRECT',RECL=IRECL,STATUS='UNKNOWN')
C
C WRITE THE INFORMATION
C
C  WRITE(NOUNIT,REC=IRDOUT)IEMAX,ALAM,THRSH,ERSHFT,IFLAV
C
C  CLOSE(NOUNIT)
C  RETURN
C  END
C
C  FUNCTION CEXRAT(ISP,ENRG,RLOC,SSN,DKTIME,IRDK,INRGDK,ISOLDK,
2    IONDK)
C
C PROGRAMMER: R. W. SPIRO
C LAST UPDATE: 05.09.90
C
C PURPOSE: FUNCTION SUBPROGRAM TO RETURN CHARGE EXCHANGE LOSS RATE
C (SEC**(-1)) FOR IONS OF SPECIES ISP, ENERGY ENRG (EV) AT
C L=RLOC (RE) FOR SUNSPOT NUMBER SSN. THIS ROUTINE IS BASED
C ON A TABLE GENERATED BY JAMES BISHOP OF U. OF MICHIGAN.
C
C CALLING PARAMETERS
C  ISP SPECIES IDENTIFIER
C  ISP=2 FOR H+ IONS
C  ISP=3 FOR O+ IONS
```

MSM79340
MSM79350
MSM79360
MSM79370
MSM79380
MSM79390
MSM79400
MSM79410
MSM79420
MSM79430
MSM79440
MSM79450
MSM79460
MSM79470
MSM79480
MSM79490
MSM79500
MSM79510
MSM79520
MSM79530
MSM79540
MSM79550
MSM79560
MSM79570
MSM79580
MSM79590
MSM79600
MSM79610
MSM79620
MSM79630
MSM79640
MSM79650
MSM79660
MSM79670
MSM79680
MSM79690
MSM79700
MSM79710
MSM79720
MSM79730
MSM79740
MSM79750
MSM79760
MSM79770
MSM79780
MSM79790
MSM79800
MSM79810
MSM79820
MSM79830
MSM79840
MSM79850
MSM79860
MSM79870
MSM79880
MSM79890
MSM79900
MSM79910
MSM79920
MSM79930
MSM79940
MSM79950
MSM79960
MSM79970

90/06/30
12:03:44

msm1.for

126

C	ENRG	ENERGY IN EV	MSM79980
C	RLOC	RADIAL LOCATION (RE)	MSM79990
C	SSN	SUNSPOT NUMBER	MSM80000
C	DKTIME	TABLE OF ION DECAY TIMES	MSM80010
C	IRDK	RADIAL DIMENSION OF DKTIME ARRAY	MSM80020
C	INRGDK	ENERGY DIMENSION OF DKTIME ARRAY	MSM80030
C	ISOLDK	SUNSPOT NUMBER DIMENSION OF DKTIME ARRAY	MSM80040
C	IONDK	NUMBER OF ION SPECIES IN DKTIME ARRAY	MSM80050
C			MSM80060
C			MSM80070
C	PARAMETER (IRSIZ=18, INRGSZ=13, ISOLSZ=2, IONSIZ=2)		MSM80080
C			MSM80090
C	DIMENSION DKTIME (IRDK, INRGDK, ISOLDK, IONDK)		MSM80100
C			MSM80110
C	DIMENSION ELGVEC (INRGSZ), RVEC (IRSIZ), SSNVEC (2)		MSM80120
C			MSM80130
C	DATA ELGVEC /2.50,2.75,3.00,3.25,3.50,3.75,4.00,		MSM80140
2	4.25,4.50,4.75,5.00,5.25,5.50/		MSM80150
C			MSM80160
C	DATA RVEC /1.50,2.00,2.50,3.00,		MSM80170
2	3.50,4.00,4.50,5.00,		MSM80180
3	5.50,6.00,6.50,7.00,		MSM80190
4	7.50,8.00,8.50,9.00,		MSM80200
5	9.50,10.00/		MSM80210
C			MSM80220
C	DATA SSNVEC /0.0,100./		MSM80230
C			MSM80240
C			MSM80250
C	CHECK DIMENSIONS		MSM80260
C			MSM80270
C	IF (IRSIZ.NE.IRDK.OR.INRGSZ.NE.INRGDK.OR.IONSIZ.NE.IONDK.OR.		MSM80280
2	ISOLSZ.NE.ISOLDK) THEN		MSM80290
	WRITE (6,*) 'DIMENSION ERROR IN SUBROUTINE CEXRAT'		MSM80300
	WRITE (6,*) 'IRDK,INRGDK,IONDK,ISOLDK',IRDK,INRGDK,IONDK,ISOLDK		MSM80310
	WRITE (6,*) 'IRSIZ,INRGSZ,IONSIZ,ISOLSZ',IRSIZ,INRGSZ,IONSIZ,		MSM80320
2	ISOLSZ		MSM80330
	WRITE (6,*) 'STOPPING PROGRAM IN CEXRAT'		MSM80340
	STOP		MSM80350
	END IF		MSM80360
C			MSM80370
C			MSM80380
C	WORK WITH LOG10 OF PARTICLE ENERGY		MSM80390
	ENRGLG=ALOG10 (ENRG)		MSM80400
C			MSM80410
C	ISPNDX=ISP-1		MSM80420
C			MSM80430
C	FIND BR FOR INTERPOLATION		MSM80440
	IF (RLOC.LE.RVEC (1)) THEN		MSM80450
	BR=1.		MSM80460
	ELSE IF (RLOC.GT.RVEC (IRDK)) THEN		MSM80470
	BR=IRDK		MSM80480
	ELSE		MSM80490
	DO 10 IR=1,IRDK-1		MSM80500
	IF (RLOC.LE.RVEC (IR+1)) THEN		MSM80510
	BR=IR+(RLOC-RVEC (IR))/(RVEC (IR+1)-RVEC (IR))		MSM80520
	GO TO 20		MSM80530
	END IF		MSM80540
10	CONTINUE		MSM80550
C			MSM80560
	END IF		MSM80570
C			MSM80580
C			MSM80590
	20 CONTINUE		MSM80600
C			MSM80610

90/06/30
12:03:44

msml.for

127

```
C
C FIND BNRG FOR INTERPOLATION
  IF (ENRGLG.LE.ELGVEC(1)) THEN
    BNRG=1.
  ELSE IF (ENRGLG.GT.ELGVEC(INRGDK)) THEN
    BNRG=INRGDK
  ELSE
    DO 30 INRG=1,INRGDK-1
      IF (ENRGLG.LE.ELGVEC(INRG+1)) THEN
        BNRG=INRG+(ENRGLG-ELGVEC(INRG))/(ELGVEC(INRG+1)-
2          ELGVEC(INRG))
        GO TO 40
      END IF
30    CONTINUE
C
  END IF
C
C
C 40 CONTINUE
C
C FIND BSSN FOR INTERPOLATION
  BSSN=1.+(SSN-SSNVEC(1))/(SSNVEC(2)-SSNVEC(1))
C
C DECAYT IS DECAY TIME IN SECONDS
  DECAYT=G3NTRP(DKTIME(1,1,1,ISPNDX),IRDK,INRGDK,ISOLDK,BR,BNRG,
2    BSSN)
C
  IF (ABS(DECAYT).LT.1.E-20) THEN
    WRITE(6,*) 'DECAYT IS LESS THAN 1.E-20 SEC IN CEXRAT'
    WRITE(6,*) 'DECAYT=',DECAYT,' BR=',BR,' BNRG=',BNRG,' BSSN=',
2    BSSN
    WRITE(6,*) 'ISP=',ISP,' ENRG=',ENRG,' RLOC=',RLOC,' SSN=',SSN
  END IF
C
C TO GET CHARGE EXCHANGE RATE (SEC**9-1) CEXRAT, INVERT DECAYT
C
  CEXRAT=1./DECAYT
C
C DEBUG 5.27.90
C   WRITE(6,800) ISP,ENRG,ENRGLG,RLOC,SSN
800  FORMAT(1X,'ISP=',I2,' ENRG=',E12.4,' ENRGLG=',F7.4,' RLOC=',
2    F7.4,' SSN=',F7.4)
C   WRITE(6,802) BR,BNRG,BSSN,DECAYT
802  FORMAT(1X,'BR=',F7.4,' BNRG=',F7.4,' BSSN=',F7.4,' DECAYT=',
2    1PE12.4)
C
C END DEBUG 5.27.90
C
C
C
C   RETURN
C   END
C   SUBROUTINE READDK(IRDK,INRGDK,ISOLDK,IONDK,DKTIME)
C
C PURPOSE:  SUBROUTINE TO READ ION CHARGE EXCHANGE DECAY TABLE
C           FURNISHED BY JAMES BISHOP OF UNIV. OF MICHIGAN
C
C LAST UPDATE: 05.10.90
C
C COMMON LUNIT CONTAINS LOGICAL UNIT NUMBERS FOR INPUT AND OUTPUT
COMMON /LUNIT/LUERR,LUPIEN,LUCORD,LUPRNT,LUIDAT,LUHDYE,
1    LUDK,LUFLXB,LUEBEG,LUSHFT,LUPIFX,LUEAVG,LUFLSM,LUFLX,
2    LUEFLX,LUEFLD,LUBFLD,LUBMIN,LUXMIN,LUYMIN,LUZMIN,LUALOC,
```

MSM80620
MSM80630
MSM80640
MSM80650
MSM80660
MSM80670
MSM80680
MSM80690
MSM80700
MSM80710
MSM80720
MSM80730
MSM80740
MSM80750
MSM80760
MSM80770
MSM80780
MSM80790
MSM80800
MSM80810
MSM80820
MSM80830
MSM80840
MSM80850
MSM80860
MSM80870
MSM80880
MSM80890
MSM80900
MSM80910
MSM80920
MSM80930
MSM80940
MSM80950
MSM80960
MSM80970
MSM80980
MSM80990
MSM81000
MSM81010
MSM81020
MSM81030
MSM81040
MSM81050
MSM81060
MSM81070
MSM81080
MSM81090
MSM81100
MSM81110
MSM81120
MSM81130
MSM81140
MSM81150
MSM81160
MSM81170
MSM81180
MSM81190
MSM81200
MSM81210
MSM81220
MSM81230
MSM81240
MSM81250

90/06/30
12:03:44

msm1.for

128

```

3      LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH
C
C      DIMENSION DKTIME (IRDK, INRGDK, ISOLDK, IONDK)
C
C      READ (LUDK, 800) DKTIME
800    FORMAT (8 (1X, 1PE9.3))
C
C      RETURN
C      END
C      SUBROUTINE FLX2ET (LATDIM, LTDIM, IEDIM, IEMAX, ITMDIM, ITM, IFLAV,
2        ALAM, ALMDEL, BNDLOC, VM, FLUX, ETA)
C
C      PURPOSE:  CONVERT FROM PARTICLE FLUX (PARTICLES/CM**2/S) TO
C                INVARIANT FLUX ETA
C
C      INPUT CALLING PARAMETERS
C      LATDIM      NUMBER OF LATITUDINAL GRID POINTS
C      LTDIM       NUMBER OF LONGITUDINAL GRID POINTS
C      IEDIM       MAXIMUM NUMBER OF MODEL ENERGY CHANNELS
C      IEMAX       ACTUAL NUMBER OF MODEL ENERGY CHANNELS
C      ITMDIM      MAXIMUM TIME DEPTH
C      ITM         TIME LABEL
C      IFLAV       VECTOR GIVING CHARGE AND MASS SPECIES OF PARTICLES
C                1=ELECTRONS, 2=H+ IONS, 3=O+ IONS
C      ALAM        ENERGY INVARIANT VECTOR
C      ALMDEL      ENERGY CHANNEL DIFFERENCE VECTOR
C      BNDLOC      BOUNDARY LOCATION ARRAY
C      VM          FLUX TUBE VOLUME**(-2/3)
C      FLUX        PARTICLE NUMBER FLUX
C
C      OUTPUT PARAMETERS
C      ETA         INVARIANT NUMBER FLUX (PARTICLES/UNIT MAGNETIC FLUX)
C
C      DIMENSION IFLAV (IEDIM), ALAM (IEDIM), ALMDEL (IEDIM)
C      DIMENSION VM (LATDIM, LTDIM, ITMDIM), FLUX (LATDIM, LTDIM, IEDIM)
C      DIMENSION ETA (LATDIM, LTDIM, IEDIM), BNDLOC (LTDIM, ITMDIM)
C
C      DO 10 IE=1, IEMAX
C        IF (IFLAV (IE).EQ.2) ATOMWT=1.
C        IF (IFLAV (IE).EQ.3) ATOMWT=16.
C
C      DO 20 J=1, LTDIM
C        IB=IFIX (BNDLOC (J, ITM))
C
C      DO 30 I=IB, LATDIM
C        IF (IFLAV (IE).EQ.1) THEN
C          ETA (I, J, IE)=FLUX (I, J, IE) *ALMDEL (IE) /7.392E-16/
2          SQRT (ABS (ALAM (IE))) /VM (I, J, ITM)
C        ELSE
C          ETA (I, J, IE)=FLUX (I, J, IE) *ALMDEL (IE) *SQRT (ATOMWT) /
2          1.731E-17/SQRT (ABS (ALAM (IE))) /VM (I, J, ITM)
C        END IF
30      CONTINUE
C
C      DO 35 I=1, IB-1
C        ETA (I, J, IE)=ETA (IB, J, IE)
35      CONTINUE
C
C      20    CONTINUE
C
C      10    CONTINUE
C
```

MSM81260
MSM81270
MSM81280
MSM81290
MSM81300
MSM81310
MSM81320
MSM81330
MSM81340
MSM81350
MSM81360
MSM81370
MSM81380
MSM81390
MSM81400
MSM81410
MSM81420
MSM81430
MSM81440
MSM81450
MSM81460
MSM81470
MSM81480
MSM81490
MSM81500
MSM81510
MSM81520
MSM81530
MSM81540
MSM81550
MSM81560
MSM81570
MSM81580
MSM81590
MSM81600
MSM81610
MSM81620
MSM81630
MSM81640
MSM81650
MSM81660
MSM81670
MSM81680
MSM81690
MSM81700
MSM81710
MSM81720
MSM81730
MSM81740
MSM81750
MSM81760
MSM81770
MSM81780
MSM81790
MSM81800
MSM81810
MSM81820
MSM81830
MSM81840
MSM81850
MSM81860
MSM81870
MSM81880
MSM81890

90/06/30
12:03:44

msm1.for

129

```
RETURN
END
SUBROUTINE ET2FLX (LATDIM, LTDIM, IEDIM, IEMAX, ITMDIM, ITM, IFLAV,
2          ALAM, ALMDEL, BNDLOC, VM, FLUX, ETA)

C
C PURPOSE:  CONVERT FROM INVARIANT FLUX ETA TO FLUX IN
C           PARTICLES/CM**2/SA
C
C INPUT CALLING PARAMETERS
C   LATDIM      NUMBER OF LATITUDINAL GRID POINTS
C   LTDIM       NUMBER OF LONGITUDINAL GRID POINTS
C   IEDIM       MAXIMUM NUMBER OF MODEL ENERGY CHANNELS
C   IEMAX       ACTUAL NUMBER OF MODEL ENERGY CHANNELS
C   ITMDIM      MAXIMUM TIME DEPTH
C   ITM         TIME LABEL
C   IFLAV       VECTOR GIVING CHARGE AND MASS SPECIES OF PARTICLES
C               1=ELECTRONS, 2=H+ IONS, 3=O+ IONS
C   ALAM        ENERGY INVARIANT VECTOR
C   ALMDEL      ENERGY CHANNEL DIFFERENCE VECTOR
C   BNDLOC      BOUNDARY LOCATION ARRAY
C   VM         FLUX TUBE VOLUME**(-2/3)
C   FLUX        PARTICLE NUMBER FLUX
C
C
C OUTPUT PARAMETERS
C   ETA         INVARIANT NUMBER FLUX (PARTICLES/UNIT MAGNETIC FLUX)
C
C
C   DIMENSION IFLAV (IEDIM), ALAM (IEDIM), ALMDEL (IEDIM)
C   DIMENSION VM (LATDIM, LTDIM, ITMDIM), FLUX (LATDIM, LTDIM, IEDIM)
C   DIMENSION ETA (LATDIM, LTDIM, IEDIM), BNDLOC (LTDIM, ITMDIM)
C
C   DO 10 IE=1, IEMAX
C     IF (IFLAV (IE) .EQ. 2) ATOMWT=1.
C     IF (IFLAV (IE) .EQ. 3) ATOMWT=16.
C
C     DO 20 J=1, LTDIM
C       IB=IFIX (BNDLOC (J, ITM))
C
C       DO 30 I=IB, LATDIM
C         IF (IFLAV (IE) .EQ. 1) THEN
C           FLUX (I, J, IE)=ETA (I, J, IE)*7.392E-16*VM (I, J, ITM)*
2             Sqrt (ABS (ALAM (IE))) /ALMDEL (IE)
C         ELSE
C           FLUX (I, J, IE)=ETA (I, J, IE)*VM (I, J, ITM)*1.731E-17*
2             Sqrt (ABS (ALAM (IE))) /ALMDEL (IE) /Sqrt (ATOMWT)
C         END IF
30      CONTINUE
C
C       DO 35 I=1, IB-1
C         FLUX (I, J, IE)=FLUX (IB, J, IE)
35      CONTINUE
C
C     20    CONTINUE
C
C     10    CONTINUE
C
C   RETURN
C   END
```

MSM81900
MSM81910
MSM81920
MSM81930
MSM81940
MSM81950
MSM81960
MSM81970
MSM81980
MSM81990
MSM82000
MSM82010
MSM82020
MSM82030
MSM82040
MSM82050
MSM82060
MSM82070
MSM82080
MSM82090
MSM82100
MSM82110
MSM82120
MSM82130
MSM82140
MSM82150
MSM82160
MSM82170
MSM82180
MSM82190
MSM82200
MSM82210
MSM82220
MSM82230
MSM82240
MSM82250
MSM82260
MSM82270
MSM82280
MSM82290
MSM82300
MSM82310
MSM82320
MSM82330
MSM82340
MSM82350
MSM82360
MSM82370
MSM82380
MSM82390
MSM82400
MSM82410
MSM82420
MSM82430
MSM82440
MSM82450
MSM82460
MSM82470
MSM82480

LIST OF B-FIELD MAT. FILES

[illegible]

BO346051.DAT;1
BO346091.DAT;1
BO346131.DAT;1
BO347011.DAT;1
BO347051.DAT;1
BO347091.DAT;1
BO347131.DAT;1
BO351011.DAT;1
BO351051.DAT;1
BO351091.DAT;1
BO351131.DAT;1
BO352011.DAT;1
BO352051.DAT;1
BO352091.DAT;1
BO352131.DAT;1
BO353011.DAT;1
BO353051.DAT;1
BO353091.DAT;1
BO353131.DAT;1
BO354011.DAT;1
BO354051.DAT;1
BO354091.DAT;1
BO354131.DAT;1
BO355011.DAT;1
BO355051.DAT;1
BO355091.DAT;1
BO355131.DAT;1
BO356011.DAT;1
BO356051.DAT;1
BO356091.DAT;1
BO356131.DAT;1
BO357011.DAT;1
BO357051.DAT;1
BO357091.DAT;1
BO357131.DAT;1
DIRECTORY.LIS;1

BO346061.DAT;1
BO346101.DAT;1
BO346141.DAT;1
BO347021.DAT;1
BO347061.DAT;1
BO347101.DAT;1
BO347141.DAT;1
BO351021.DAT;1
BO351061.DAT;1
BO351101.DAT;1
BO351141.DAT;1
BO352021.DAT;1
BO352061.DAT;1
BO352101.DAT;1
BO352141.DAT;1
BO353021.DAT;1
BO353061.DAT;1
BO353101.DAT;1
BO353141.DAT;1
BO354021.DAT;1
BO354061.DAT;1
BO354101.DAT;1
BO354141.DAT;1
BO355021.DAT;1
BO355061.DAT;1
BO355101.DAT;1
BO355141.DAT;1
BO356021.DAT;1
BO356061.DAT;1
BO356101.DAT;1
BO356141.DAT;1
BO357021.DAT;1
BO357061.DAT;1
BO357101.DAT;1
BO357141.DAT;1

BO346071.DAT;1
BO346111.DAT;1
BO346151.DAT;1
BO347031.DAT;1
BO347071.DAT;1
BO347111.DAT;1
BO347151.DAT;1
BO351031.DAT;1
BO351071.DAT;1
BO351111.DAT;1
BO351151.DAT;1
BO352031.DAT;1
BO352071.DAT;1
BO352111.DAT;1
BO352151.DAT;1
BO353031.DAT;1
BO353071.DAT;1
BO353111.DAT;1
BO353151.DAT;1
BO354031.DAT;1
BO354071.DAT;1
BO354111.DAT;1
BO354151.DAT;1
BO355031.DAT;1
BO355071.DAT;1
BO355111.DAT;1
BO355151.DAT;1
BO356031.DAT;1
BO356071.DAT;1
BO356111.DAT;1
BO356151.DAT;1
BO357031.DAT;1
BO357071.DAT;1
BO357111.DAT;1
BO357151.DAT;1

BO346081.DAT;1
BO346121.DAT;1
BO346161.DAT;1
BO347041.DAT;1
BO347081.DAT;1
BO347121.DAT;1
BO347161.DAT;1
BO351041.DAT;1
BO351081.DAT;1
BO351121.DAT;1
BO351161.DAT;1
BO352041.DAT;1
BO352081.DAT;1
BO352121.DAT;1
BO352161.DAT;1
BO353041.DAT;1
BO353081.DAT;1
BO353121.DAT;1
BO353161.DAT;1
BO354041.DAT;1
BO354081.DAT;1
BO354121.DAT;1
BO354161.DAT;1
BO355041.DAT;1
BO355081.DAT;1
BO355121.DAT;1
BO355161.DAT;1
BO356041.DAT;1
BO356081.DAT;1
BO356121.DAT;1
BO356161.DAT;1
BO357041.DAT;1
BO357081.DAT;1
BO357121.DAT;1
BO357161.DAT;1

Total of 609 files.

Appendix C

Appendix C is attached separately as a computer listing.

Appendix D

90/06/30
12:04:50

mapint.com

1

```
$define sys$input tt
!$define sys$output tt
$define bfield [-.bfields]
$define for008 msmout.dat
$define for014 ashift.dat
$define for015 aflux.dat
$define for016 avm.dat
$define for017 abndloc.dat
$define for018 axmin.dat
$define for019 aymin.dat
$define for020 azmin.dat
$define for044 aupdat.dat
$run mapint
```

90/06/30
12:04:14

mapint.for

1

```
PROGRAM MSMAP                                MAP00010
C      VERSION OF 4/23/90                      MAP00020
C      PROGRAMMED BY R. V. HILMER              MAP00030
C      REQUIRES LINKAGE WITH BFIELD MODEL SUBROUTINES, MAP00040
C      SUBROUTINES FNDBRK, AND B-MATRIX DATA FILES MAP00050
C                                              MAP00060
C      PURPOSE- THIS PROGRAM COLLECTS MAGNETOSPHERIC SPECIFICATION MAP00070
C      MODEL INPUTS AND AN ARBITRARY GSM STARTING POINT, MAP00080
C      VALIDATES THE INPUTS AND FINDS THE POINT ON THE MSM MAP00090
C      CALCULATION SURFACE WHICH IS MAGNETICALLY CONNECTED MAP00100
C      TO THE GSM STARTING POINT. MAP00110
C      IT THEN INTERPOLATES IN SPACE, TIME AND ENERGY TO MAP00120
C      GET THE FLUXES AT THE MAPPED POINT. MAP00130
C      VARIABLES- FTILT: DIPOLE TILT ANGLE (DEGREES, POSITIVE FOR MAP00140
C      NORTHERN HEMISPHERE SUMMER) MAP00150
C      FSTDDF: MAGNETOPAUSE STAND-OFF DISTANCE (RE) MAP00160
C      FDST: MAGNETIC ACTIVITY INDEX DST (NT) MAP00170
C      FEQEDG: DIPOLE LAT. OF MIDNIGHT EQUATORWARD DIFFUSE MAP00180
C      AURORA BOUNDARY (DEG). MAP00190
C      FCLPSE: MAGNETOTAIL FIELD COLLAPSE INDICATOR MAP00200
C      XSTRT,YSTRT,ZSTRT: MAGNETIC TRACING START POINT (GSM) MAP00210
C      IVALID: INDICATES IF INSERTED INPUTS ARE WITHIN VALID MAP00220
C      RANGES. (1 = YES, 0 = NO) MAP00230
C      XMAP,YMAP,ZMAP: MAGNETIC MAPPING POINT ON THE MSM MAP00240
C      COMPUTATIONAL SURFACE (GSM). MAP00250
C      SUBROUTINES-MAPTRK: DETERMINES THE MAGNETIC FIELD MAPPING MAP00260
C      CONNECTING AN ARBITRARY POINT IN SPACE TO THE MSM MAP00270
C      COMPUTATIONAL SURFACE. USES INTERPOLATION SCHEME MAP00280
C      INVOLVING PRECALCULATED INTERNAL MODEL PARAMETER MAP00290
C      SETS TO FIND THE MAPPING FOR ARBITRARY MSM INPUT. MAP00300
C      VALID: CHECKS TO SEE IF EXTERNAL INPUT PARAMETERS MAP00310
C      AND TRACING START POINT ARE VALID. MAP00320
C      READ3D: READS DATA FILES FOR FLUX CALCULATION. MAP00330
C      LOCATE: FINDS THE INTERPOLATED POINT ON A MAP00340
C      2-DIMENSIONAL GRID. MAP00350
C      CHECK: CHECKS THAT THE INPUT TIMES ARE REASONABLE. MAP00360
C      TCHECK: LOCATES THE 2 RECORDS WHICH SPAN THE TIME MAP00370
C      REQUESTED. MAP00380
C      RDVEC: READS THE HEADER VECTOR OF THE FLUX FILES MAP00390
C      SETALM: SETS UP THE INVARIANT ENERGY ARRAY MAP00400
C      FUNCTIONS: G3NTRP: 3-DIMENSIONAL INTERPOLATION ROUTINE MAP00410
C
C      PARAMETER (LATDIM=62,LTDIM=51,IEDIM=30,ITMDIM=50, NAUGEL=28) MAP00420
C      DIMENSION R(LATDIM,LTDIM),BNDLOC(LTDIM,2),P(LATDIM,LTDIM), MAP00430
1  VM(LATDIM,LTDIM,2),FLUX(LATDIM,LTDIM,IEDIM),ALAM(IEDIM), MAP00440
2  ALMDEL(IEDIM),ID(20),RID(20),THRSH(IEDIM),ERSHFT(IEDIM), MAP00450
3  XMIN(LATDIM,LTDIM,2),YMIN(LATDIM,LTDIM,2),ZMIN(LATDIM,LTDIM,2), MAP00460
4  AUGPAR(NAUGEL),AGPAR1(NAUGEL),FLUX1(LATDIM,LTDIM,IEDIM), MAP00470
5  ISTART(3),IFLAV(IEDIM),FLXOUT(2) MAP00480
C      CHARACTER*80 CHID MAP00490
C
C      PI=ATAN2(0.,-1.) MAP00500
C
C      THE INTERACTIVE PORTION OF THE PROGRAM CHECKS FOR CORRECT TIME MAP00510
C      ENTRIES THROUGH SUBROUTINE CHECK. MAP00520
C
C      MAP00530
C      MAP00540
100 WRITE(6,*)' ENTER LAST 2 DIGITS OF YEAR AND JULIAN DAY ' MAP00550
C      READ(5,*)IYEAR,IDAY MAP00560
C      WRITE(6,*)' ENTER HOUR, MINUTE, AND SECONDS ' MAP00570
C      READ(5,*)Ihour,MINUTE,ISCNDS MAP00580
C      CALL CHECK(IYEAR,IDAY,Ihour,MINUTE,ISCNDS,IFLAG) MAP00590
C      IF(IFLAG.EQ. 1)THEN MAP00600
C      WRITE(6,*) MAP00610
C      WRITE(6,*)' INCORRECT INPUT. DO YOU WANT TO TRY AGAIN?' MAP00620
```

90/06/30
12:04:14

3

mapint.for

```
C      WRITE(6,*) 'ALAM', ALAM, 'THRSH', THRSH, 'ERSHFT', ERSHT, 'IFLAV', IFLAV

      CALL READ3D(15, 'FLUX ', IREC, LATDIM, LTDIM, IEMAX, ITMDIM, ID, RID,
1      CHID, FLUX)
      CALL READ3D(15, 'FLUX ', IREC+1, LATDIM, LTDIM, IEMAX, ITMDIM, ID,
1      RID, CHID, FLUX1)

      IF ((ID(8).NE.LATDIM).OR.(ID(9).NE.LTDIM)) THEN
      WRITE(6,*)
      WRITE(6,*) ' INCORRECT PARAMETERS FOR GRID-STOPPING PROGRAM'
      WRITE(6,*) ' ID(8) = ', ID(8), ' LATDIM = ', LATDIM
      WRITE(6,*) ' ID(9) = ', ID(9), ' LTDIM = ', LTDIM
      WRITE(6,*)
      STOP
      ENDIF
      CALL READ3D(16, 'VM      ', IREC, LATDIM, LTDIM, 1, ITMDIM, ID, RID, CHID,
1      VM(1,1,1))
      CALL READ3D(16, 'VM      ', IREC+1, LATDIM, LTDIM, 1, ITMDIM, ID, RID, CHID,
1      VM(1,1,2))
      CALL READ3D(17, 'BNDLOC', IREC, LTDIM, 1, 1, ITMDIM, ID, RID, CHID,
1      BNDLOC(1,1))
      CALL READ3D(17, 'BNDLOC', IREC+1, LTDIM, 1, 1, ITMDIM, ID, RID, CHID,
1      BNDLOC(1,2))
      CALL READ3D(18, 'XMIN  ', IREC, LATDIM, LTDIM, 1, ITMDIM, ID, RID,
1      CHID, XMIN(1,1,1))
      CALL READ3D(18, 'XMIN  ', IREC+1, LATDIM, LTDIM, 1, ITMDIM, ID, RID,
1      CHID, XMIN(1,1,2))
      CALL READ3D(19, 'YMIN  ', IREC, LATDIM, LTDIM, 1, ITMDIM, ID, RID,
1      CHID, YMIN(1,1,1))
      CALL READ3D(19, 'YMIN  ', IREC+1, LATDIM, LTDIM, 1, ITMDIM, ID, RID,
1      CHID, YMIN(1,1,2))
      CALL READ3D(20, 'ZMIN  ', IREC, LATDIM, LTDIM, 1, ITMDIM, ID, RID,
1      CHID, ZMIN(1,1,1))
      CALL READ3D(20, 'ZMIN  ', IREC+1, LATDIM, LTDIM, 1, ITMDIM, ID, RID,
1      CHID, ZMIN(1,1,2))

C
C      SET UP THE TIME INTERPOLATION FOR THE B-FIELD COMPONENTS AND
C      CALCULATE RADIUS AND LOCAL TIME ARRAYS
C
      TIME=(FLOAT(IHOUR)*3600.0) + (FLOAT(MINUTE)*60.0) + FLOAT(ISCNDS)
      TIME1=(AUGPAR(3)*3600.0) + (AUGPAR(4)*60.0) + AUGPAR(5)
      TIME2=(AGPAR1(3)*3600.0) + (AGPAR1(4)*60.0) + AGPAR1(5)
      TIMTRP=((TIME-TIME1)/(TIME2-TIME1)) + 1.0
C      WRITE(6,*) 'TIME, TIME1, TIME2, TIMTRP ', TIME, TIME1, TIME2, TIMTRP
      DO 400 I=1, LATDIM
      DO 450 J=1, LTDIM
      XNTRP=G3NTRP(XMIN, LATDIM, LTDIM, 2, FLOAT(I), FLOAT(J), TIMTRP)
      YNTRP=G3NTRP(YMIN, LATDIM, LTDIM, 2, FLOAT(I), FLOAT(J), TIMTRP)
      ZNTRP=G3NTRP(ZMIN, LATDIM, LTDIM, 2, FLOAT(I), FLOAT(J), TIMTRP)
      R(I,J)=SQRT(XNTRP**2 + YNTRP**2 + ZNTRP**2)
      IF ((XNTRP.EQ.0.0).AND.(YNTRP.EQ.0.0)) THEN
      P(I,J)=0.0
      ELSE
      P(I,J)=ATAN2(YNTRP,XNTRP)
      IF (P(I,J).LT.0.0) P(I,J)=P(I,J)+ 2.0*PI
      ENDIF
450    CONTINUE
400    CONTINUE
C      WRITE(6,*) ' IREC ', IREC
C
C      SET UP INVARIANT ARRAYS FOR THE FLUX CALCULATIONS
C
      CALL SETALM(ALAM, IEMAX, IEDIM, IFLAV, ALMDEL)
10    CONTINUE
```

90/06/30
12:04:14

mapint.for

2

```
WRITE(6,*)' ENTER 0 TO STOP, ANY OTHER NUMBER TO TRY AGAIN.'
```

```
READ(5,*) IANSW
```

```
WRITE(6,*)
```

```
IF(IANSW.EQ.0) STOP
```

```
GO TO 100
```

```
ENDIF
```

```
IREC=1
```

```
C
```

```
C THIS PORTION OF THE CODE SEARCHES THE OUTPUT DATA BASE TO FIND THE
```

```
C 2 TIMES CLOSEST TO THE INPUT TIME.
```

```
C
```

```
200 CALL READ3D(44,'UPDATE',IREC,NAUGEL,1,1,ITMDIM,ID,RID,CHID,
```

```
1 AUGPAR)
```

```
C WRITE(6,*)' IREC, ID ',IREC,ID
```

```
IF(IREC.EQ.1) THEN
```

```
ISTART(1)=AUGPAR(1)
```

```
ISTART(2)=AUGPAR(2)
```

```
ISTART(3)=AUGPAR(3)*3600 + AUGPAR(4)*60 + AUGPAR(5)
```

```
ITMMAX=ID(10)
```

```
IEMAX=ID(9)
```

```
ENDIF
```

```
IF(IREC+1.GT.ITMMAX) THEN
```

```
WRITE(6,*)
```

```
WRITE(6,*)' END OF FILE ENCOUNTERED WITHOUT FINDING THE TIME',
```

```
1 ' YOU REQUESTED.'
```

```
WRITE(6,*)' ENTER 0 TO STOP, ANY OTHER NUMBER TO TRY AGAIN.'
```

```
READ(5,*) IANSW
```

```
IF(IANSW.EQ.0) STOP
```

```
GO TO 100
```

```
ENDIF
```

```
CALL READ3D(44,'UPDATE',IREC+1,NAUGEL,1,1,ITMDIM,ID,RID,CHID,
```

```
1 AGPAR1)
```

```
CALL TCHECK(IYEAR,IDAY,IHOUR,MINUTE,ISCNDS,AUGPAR,AGPAR1,
```

```
1 NAUGEL,IFLAG,ISTART)
```

```
IF(IFLAG.EQ.1) THEN
```

```
IREC=IREC+1
```

```
GO TO 200
```

```
ENDIF
```

```
C
```

```
C INPUT SATELLITE ENERGY IN KEV. THIS IS CHECK TO BE GREATER THAN 0
```

```
C AND CONVERTED TO INVARIANT ENERGY, ALAM, THE PROGRAMS BASIC
```

```
C ENERGY UNIT. ALSO PARTICLE TYPE IS REQUESTED AND CHECKED TO BE
```

```
C BETWEEN 1 AND 3 (1=ELECTRONS,2=H+,3=O+)
```

```
C
```

```
300 WRITE(6,*)' INPUT ENERGY (KEV) AND 1 FOR ELECTRONS, 2 FOR H+,'
```

```
WRITE(6,*)' OR 3 FOR O+'
```

```
READ(5,*) ENERGY,JFLAV
```

```
IF((JFLAV.LT.1).OR.(JFLAV.GT.3)) THEN
```

```
WRITE(6,*)
```

```
WRITE(6,*)' INCORRECT INPUT FOR PARTICLE TYPE. PLEASE RE-ENTER.'
```

```
WRITE(6,*)
```

```
GO TO 300
```

```
ENDIF
```

```
IF(ENERGY.LE.0) THEN
```

```
WRITE(6,*)
```

```
WRITE(6,*)' ENERGY LESS THAN 0. PLEASE RE-ENTER.'
```

```
WRITE(6,*)
```

```
GO TO 300
```

```
ENDIF
```

```
C
```

```
C READ INPUT FILES NEEDED FOR INTERPOLATING
```

```
C
```

```
CALL RDVEC (14,IREC,IEMAX,IEDIM,ALAM,THRSH,ERSHFT,IFLAV)
```


90/06/30
12:04:14

mapint.for

5

```

550  ENDIF
      CONTINUE
      IF (XLO.EQ.-1.0) THEN
        WRITE(6,*)
        WRITE(6,*) ' ENERGY NOT CALCULATED FOR THIS POINT IN THIS RUN.'
        WRITE(6,*)
        WRITE(6,*) ' FOLLOWING ARE THE ENERGIES AND SPECIES FOR THIS RUN'
        WRITE(6,*)
        DO 475 IE=1,IEMAX
          WRITE(6,*) ' ENERGY (KEV) = ',ABS(ALAM(IE)*CALVM)/1000.0,
1          ' SPECIES NUMBER = ',IFLAV(IE)
475  CONTINUE
        WRITE(6,*) ' ENTER 0 TO STOP, ANY OTHER NUMBER TO TRY AGAIN'
        READ(5,*) IANSW
        IF (IANSW.EQ.0) STOP
        WRITE(6,*)
        GO TO 300
      ENDIF
      IF ((ALAM(IFIX(BK)).LT.0.0).AND.(JFLAV.NE.1)) THEN
        WRITE(6,*)
        WRITE(6,*) ' PARTICLE TYPE AND INVARIANT ENERGY ARE INCORRECT.'
        WRITE(6,*) ' ENTER 0 STOP, AND OTHER NUMBER TO CONTINUE.'
        READ(5,*) IANSW
        WRITE(6,*)
        IF (IANSW.EQ.0) STOP
        GO TO 300
      ENDIF
C
C  INTERPOLATE FLUX ARRAYS IN SPACE, TIME AND ENERGY
C
C
      FLXOUT(1)=G3NTRP (FLUX,LATDIM,LTDIM,IEMAX,BI,BJ,BK)
      FLXOUT(2)=G3NTRP (FLUX1,LATDIM,LTDIM,IEMAX,BI,BJ,BK)
      SATFLX=G3NTRP (FLXOUT,2,1,1,TIMTRP,1.,1.)
C      WRITE(6,*) ' BI,BJ,BK,TIMTRP,FLXOUT ',BI,BJ,BK,TIMTRP,FLXOUT
C
C  WRITE OUT FLUX, INCLUDING ERROR AND THRESHOLD IF AVAILABLE
C
      ERROR=G3NTRP (ERSHFT,IEMAX,1,1,BK,1.,1.)
      WRITE(6,*)
      WRITE(6,*) ' FLUX AT THE SATELLITE (LOG(/CM2-S-KEV-STER)) = ',
1      SATFLX
      WRITE(6,*)
      IF (ERROR.NE.0.0) THEN
        WRITE(6,*) ' THE ERROR BETWEEN THE MODEL OUTPUT AND SATELLITE '
        WRITE(6,*) ' DATA IS ',ERROR
        WRITE(6,*) ' PLEASE NOTE THAT THE ERROR WAS CALCULATED USING '
        WRITE(6,*) ' DATA AT GEOSYNCHRONOUS ORBIT AND IS ONLY VALID FOR'
        WRITE(6,*) ' POINTS AT 6.6 RE.'
        WRITE(6,*)
      ENDIF
      THRSHK=G3NTRP (THRSH,IEMAX,1,1,BK,1.,1.)
      WRITE(6,*)
      IF (THRSHK.NE.0.0) THEN
        WRITE(6,*) ' THE PARTICLES USED IN THE FLUX CALCULATION WERE '
        WRITE(6,*) ' ABOVE THE THRESHOLD LEVEL AND THE FULL-TRACEBACK'
        WRITE(6,*) ' METHOD WAS NOT USED.'
      ENDIF
C      WRITE(6,*) ' ERROR, THRSHK ',ERROR,THRSHK
      WRITE(6,*)
500  WRITE(6,*) ' TRACE FROM ANOTHER STARTING POINT'
      WRITE(6,*) ' WITH SAME TIME PARAMETERS? (1 = YES)'
      READ(5,*) KK
      IF (KK.EQ.1) GOTO 300
```

MAP02520
MAP02530
MAP02540
MAP02550
MAP02560
MAP02570
MAP02580
MAP02590
MAP02600
MAP02610
MAP02620
MAP02630
MAP02640
MAP02650
MAP02660
MAP02670
MAP02680
MAP02690
MAP02700
MAP02710
MAP02720
MAP02730
MAP02740
MAP02750
MAP02760
MAP02770
MAP02780
MAP02790
MAP02800
MAP02810
MAP02820
MAP02830
MAP02840
MAP02850
MAP02860
MAP02870
MAP02880
MAP02890
MAP02900
MAP02910
MAP02920
MAP02930
MAP02940
MAP02950
MAP02960
MAP02970
MAP02980
MAP02990
MAP03000
MAP03010
MAP03020
MAP03030
MAP03040
MAP03050
MAP03060
MAP03070
MAP03080
MAP03090
MAP03100
MAP03110
MAP03120
MAP03130
MAP03140
MAP03150

90/06/30
12:04:14

mapint.for

4

```
C
C INPUT FOR SPATIAL COORDINATE OF SATELLITE
C
    FTILT= AUGPAR(23)
    FSTDFF=AUGPAR(24)
    FDST=AUGPAR(7)
    FEQEDG=AUGPAR(8)
    FCLPSE=AUGPAR(15)
20  CONTINUE
    WRITE(6,*) 'ENTER SPATIAL POINT OF INTEREST IN GSM (RE): X,Y,Z'
    READ(5,*) XSTRT,YSTRT,ZSTRT
C    CHECK VALIDITY OF INPUT PARAMETERS AND TRACER START POINT
    CALL VALID(FTILT,FSTDFF,FDST,FEQEDG,FCLPSE,XSTRT,YSTRT,ZSTRT,
1      IVALID)
    IF (IVALID.EQ.0) THEN
        WRITE(6,*) 'INVALID INPUT. TRY AGAIN? (0 = YES)'
        READ(5,*) KK
        IF (KK.EQ.0) GOTO 20
        STOP
    END IF
    CALL MAPTRK(FSTDFF,FTILT,FEQEDG,FDST,FCLPSE,XSTRT,YSTRT,ZSTRT,
1      XMAP,YMAP,ZMAP)
    IF (ABS(XMAP).LT.1.0.AND.ABS(YMAP).LT.1.0.AND.ABS(ZMAP).LT.1.0)
1  GO TO 500
    WRITE(6,*)
    WRITE(6,*) 'INTERPOLATED MAPPING POINT: XGSM   YGSM   ZGSM'
    WRITE(6,*) XMAP,YMAP,ZMAP
    WRITE(6,*)
C
C CHECK THAT THE MAPPED POINT IS WITHIN THE BOUNDARY OF THE MODEL
C CALCULATION
C
    RTEST=SQRT(XMAP**2 + YMAP**2 + ZMAP**2)
    PTEST=ATAN2(YMAP,XMAP)
    IF(PTEST.GT.2.0*PI)PTEST=PTEST-(2.0*PI)
    IF(PTEST.LT.0.0)PTEST=PTEST + 2.0*PI
    CALL LOCATE(RTEST,PTEST,LATDIM,LTDIM,R,P,BI,BJ)
    JPLACE=IFIX(BJ)
    BNDY=G3NTRP(BNDLOC,LTDIM,2,1,BJ,TIMTRP,1.0)
    IPLACE=IFIX(BNDY)
    IF((R(IPLACE,JPLACE).LT.RTEST)) THEN
        WRITE(6,*)
        WRITE(6,*) 'POINT MAPPED TO IS OUT OF THE BOUNDARY OF THE '
        WRITE(6,*) ' MSM CALCULATION REGION. '
        WRITE(6,*) ' BOUNDARY LOCATION ',R(IPLACE,JPLACE),' RE'
        WRITE(6,*) ' SATELLITE MAPS TO ',RTEST,' RE'
        WRITE(6,*)
        GO TO 500
    ENDIF
    CALVM=G3NTRP(VM,LATDIM,LTDIM,2,BI,BJ,TIMTRP)
    ALMCHK=(ENERGY/CALVM)*1000.0
C    WRITE(6,*) 'ENERGY,CALVM,ALMCHK',ENERGY,CALVM,ALMCHK
C    WRITE(6,*) ' VM(IFIX(BI,BJ)) ',VM(IFIX(BI),IFIX(BJ),1)
C
C CHECK TO SEE IF THE ENERGY AND SPECIES INPUT IS WITHIN OUR
C CALCULATED RANGE
C
    XLO=-1.0
    DO 550 IE=1,IEMAX-1
        IF((IFLAV(IE).NE.JFLAV).AND.(IFLAV(IE+1).NE.JFLAV))GO TO 550
        IF((ALMCHK.GE.ABS(ALAM(IE))).AND.(ALMCHK.LE.ABS(ALAM(IE+1))))
1  .AND.(IFLAV(IE).EQ.IFLAV(IE+1))) THEN
            XLO=(ALMCHK-ABS(ALAM(IE)))/(ABS(ALAM(IE+1))-ABS(ALAM(IE)))
            BK=FLOAT(IE)+XLO
MAP01880
MAP01890
MAP01900
MAP01910
MAP01920
MAP01930
MAP01940
MAP01950
MAP01960
MAP01970
MAP01980
MAP01990
MAP02000
MAP02010
MAP02020
MAP02030
MAP02040
MAP02050
MAP02060
MAP02070
MAP02080
MAP02090
MAP02100
MAP02110
MAP02120
MAP02130
MAP02140
MAP02150
MAP02160
MAP02170
MAP02180
MAP02190
MAP02200
MAP02210
MAP02220
MAP02230
MAP02240
MAP02250
MAP02260
MAP02270
MAP02280
MAP02290
MAP02300
MAP02310
MAP02320
MAP02330
MAP02340
MAP02350
MAP02360
MAP02370
MAP02380
MAP02390
MAP02400
MAP02410
MAP02420
MAP02430
MAP02440
MAP02450
MAP02460
MAP02470
MAP02480
MAP02490
MAP02500
MAP02510
```

```

C      PURPOSE  -DETERMINE THE MAGNETIC FIELD MAPPING CONNECTING AN      MAP03800
C      ARBITRARY POINT IN SPACE TO THE MSM COMPUTATIONAL                MAP03810
C      SURFACE (THE GSM EQUATORIAL PLANE FOR DIPOLE TILT ANGLEMAP03830
C      SET EQUAL TO ZERO) MAPPINGS ARE DONE FROM A GIVEN POINTMAP03840
C      TO A SURFACE FOR ALL MODELS IMMEDIATELY ADJACENT IN      MAP03850
C      THE MODELS EXTERNAL PARAMETER SPACE. (SEE OUTLINE OF      MAP03860
C      METHOD GIVEN IN MSM QUARTERLY REPORT NO. 11)              MAP03870
C      *** NOTE: PROCEDURE IS STOPPED IF AN OPEN FIELD-LINE IS FOUND      MAP03880
C      DURING ONE OF THE TRACES OR IF THERE ARE LESS THAN THE      MAP03890
C      MINIMUM NUMBER (8) OF ESTABLISHED MODELS REQUIRED FOR      MAP03900
C      A PROPER INTERPOLATION.                                     MAP03910
C      INPUT    -FTILT: DIPOLE TILT ANGLE (DEGREES, POSITIVE FOR      MAP03920
C      NORTHERN HEMISPHERE SUMMER.                                MAP03930
C      FSTDFF: MAGNETOPAUSE STAND-OFF DISTANCE (RE)              MAP03940
C      FDST: MAGNETIC ACTIVITY DST (NT)                           MAP03950
C      FEQEDG: DIPOLE LAT. OF MIDNIGHT EQUATORWARD DIFFUSE      MAP03960
C      AURORA BOUNDARY (DEG)                                     MAP03970
C      FCLPSE: MAGNETOTAIL FIELD COLLAPSE INDICATOR             MAP03980
C      XSTRT,YSTRT,ZSTRT: STARTING POINT FOR MAGNETIC FIELD      MAP03990
C      TRACES (GSM COORDINATES)                                  MAP04000
C      OUTPUT   -XMAP,YMAP,ZMAP: INTERPOLATED MAPPING POSITION (GSM)  MAP04010
C      VARIABLES -XHIT,YHIT,ZHIT: MAPPING POINT OF INDIVIDUAL TRACES  MAP04020
C      IOPEN: OPEN/CLOSED FIELD-LINE INDICATOR 1 = OPEN          MAP04030
C      0 = CLOSED                                                MAP04040
C      BFEKST: INDICATES IF B-FIELD MATRIX EXISTS               MAP04050
C      TPRM: MAGNETIC FIELD MODEL INTERNAL INPUT PARAMETERS      MAP04060
C      BFLIM: ESTABLISHED PARAMETERS WHICH BRACKET BFPAR        MAP04070
C      T: COEFFICIENTS FOR SINGLE LINEAR INTERPOLATION USED      MAP04080
C      CONSTRUCT VALUES OF VARIABLE COEFF.                     MAP04090
C      BFPAR: THE INPUT PARAMETERS                               MAP04100
C      COEFF: INTERPOLATION COEFFICIENT                          MAP04110
C      ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,COLLAP,DELXC,MAP04120
C      DYC,BRN,BRP,RN,RP: MAGNETIC FIELD MODEL INTERNAL          MAP04130
C      INPUTS (SEE SUBROUTINE BFIELD FOR DETAILS)                 MAP04140
C      XMAX,YMAX,ZMAX: MAXIMUM VALUES OF XHIT,YHIT,ZHIT        MAP04150
C      XMIN,YMIN,ZMIN: MINIMUM VALUES OF XHIT,YHIT,ZHIT        MAP04160
C      XRANGE,YRANGE,ZRANGE: = MAXIMUM - MINIMUM OF ABOVE       MAP04170
C      SUBROUTINE-BFIELD: MAGNETIC FIELD MODEL POINT CALCULATIONS MAP04180
C      BMNMAP: MAPS SINGLE FIELD LINE TO MINIMUM |B| SURFACE    MAP04190
C      GETIP: COLLECTS INTERNAL BFIELD MODEL INPUTS             MAP04200
C      FROM THE RESPECTIVE MATRICES FOR THE GROUP               MAP04210
C      OF MODELS IN CLOSEST PROXIMITY OF PARAMETER              MAP04220
C      SPACE TO THE MSM INPUTS.                                  MAP04230
C      MAP04240
C      NOTE: THIS SUBROUTINE CONTAINS PARTS WHICH WERE ADAPTED DIRECTLY      MAP04250
C      FROM SUBROUTINE BTRACE AS PROGRAMMED BY R.W. SPIRO. THESE      MAP04260
C      INCLUDE THE MATRIX READING PROCEDURE, DEVELOPING THE PARAMETER      MAP04270
C      WEIGHTING COEFFICIENTS, AND THE FIVE DIMENSIONAL INTERPOLATION      MAP04280
C      SCHEME.                                                    MAP04290
C      REFERENCE:                                                 MAP04300
C      FIVE DIMENSIONAL INTERPOLATION IS BASED ON MULTILINEAR      MAP04310
C      INTERPOLATION SCHEME GIVEN BY PRESS ET AL. IN             MAP04320
C      NUMERICAL RECIPES, CAMBRIDGE UNIVERSITY PRESS, 1987,PP 95-97      MAP04330
C      MAP04340
C      LOGICAL*1 BFEKST(2,2,2,2,2)                                MAP04350
C      MAP04360
C      DIMENSION TPRM(14,2,2,2,2,2)                                MAP04370
C      DIMENSION XHIT(2,2,2,2,2),YHIT(2,2,2,2,2),ZHIT(2,2,2,2,2)      MAP04380
C      DIMENSION BFLIM(2,5),BFPAR(5)                                MAP04390
C      DIMENSION T(2,5),N(5)                                        MAP04400
C      MAP04410
C      SET UP BFPAR VECTOR FOR CALLING GETMAT                     MAP04420
C      BFPAR(1)=FSTDFF                                             MAP04430

```

90/06/30
12:04:14

6

mapint.for

```
WRITE(6,*) 'AGAIN WITH NEW TIME?' (1 = YES)'
READ(5,*) KK
IF (KK.EQ.1) GOTO 100

C
STOP
END

C
C***** VALID
SUBROUTINE VALID(TILT,STDF,DST,EQBDY,COLLAP,XSTRT,YSTRT,ZSTRT,
1 IVALID)

C
C PURPOSE- DETERMINE IF EXTERIOR MODEL INPUTS ARE WITHIN VALID
C RANGES.
C INPUT- TILT: DIPOLE TILT ANGLE (DEG)
C STDF: MAGNETOPAUSE STAND-OFF DISTANCE (RE)
C DST: MAGNETIC FIELD STRENGTH INDEX (NT)
C EQBDY: MIDNIGHT EQUATORWARD BOUNDARY OF DIFFUSE
C AURORA (DEGREES DIPOLE LATITUDE)
C COLLAP: SUBSTORM MAGNETOTAIL FIELD COLLAPSE INDICATOR
C XSTRT,YSTRT,ZSTRT: GSM STARTING POINT FOR B-FIELD TRACE
C OUTPUT- IVALID: INDICATOR OF PARAMETER SET VALIDITY
C 1 = VALID, 0 = INVALID
C VARIABLE-DIST: GEOCENTRIC DISTANCE TO TRACER STARTING POINT (RE)
C
C SET MAXIMUM(MX) AND MINIMUM(MN) ACCEPTABLE PARAMETER VALUES
DATA TILTMX,TILTMN/35.00,-35.00/
DATA STDFMX,STDFMN/14.00,6.00/
DATA DSTMX,DSTMN/50.00,-400.00/
DATA EQBDYMX,EQBDYMN/69.30,49.47/
DATA COLLAPMX,COLLAPMN/1.00,0.00/

C
IVALID = 1
IF ((TILT.LT.TILTMN).OR.(TILT.GT.TILTMX)) THEN
IVALID = 0
WRITE(6,*) 'TILT VALUE',TILT,'IS OUT OF RANGE'
END IF
IF ((STDF.LT.STDFMN).OR.(STDF.GT.STDFMX)) THEN
IVALID = 0
WRITE(6,*) 'STAND-OFF VALUE',STDF,'IS OUT OF RANGE'
END IF
IF ((DST.LT.DSTMN).OR.(DST.GT.DSTMX)) THEN
IVALID = 0
WRITE(6,*) 'DST VALUE',DST,'IS OUT OF RANGE'
END IF
IF ((EQBDY.LT.EQBDYMN).OR.(EQBDY.GT.EQBDYMX)) THEN
IVALID = 0
WRITE(6,*) 'EQEDGE VALUE',EQBDY,'IS OUT OF RANGE'
END IF
IF ((COLLAP.LT.COLLAPMN).OR.(COLLAP.GT.COLLAPMX)) THEN
IVALID = 0
WRITE(6,*) 'COLLAPSE VALUE',COLLAP,'IS OUT OF RANGE'
END IF

C
DIST = SQRT(XSTRT**2 + YSTRT**2 + ZSTRT**2)
IF (DIST.LE.1.) THEN
IVALID = 0
WRITE(6,*) 'INVALID STARTING POINT FOR FIELD-LINE TRACE (<1 RE)'
END IF

C
RETURN
END

C***** MAPTRK
SUBROUTINE MAPTRK(FSTDF,FTILT,FEQEDG,FDST,FCLPSE,
1 XSTRT,YSTRT,ZSTRT,XMAP,YMAP,ZMAP)
```

90/06/30
12:04:14

mapint.for

8

```
BFPAR(2)=FTILT
BFPAR(3)=FEQEDG
BFPAR(4)=FDST
BFPAR(5)=FCLPSE
C
C INITIALIZE "HIT" ARRAYS AND MAPPING RESULT
  X = 1.
  Y = 1.
  Z = 1.
  XMAP = 0.
  YMAP = 0.
  ZMAP = 0.
  DO 15 I=1,2
    DO 16 J=1,2
      DO 17 K=1,2
        DO 18 L=1,2
          DO 19 M=1,2
            XHIT(I,J,K,L,M) = 0.
            YHIT(I,J,K,L,M) = 0.
            ZHIT(I,J,K,L,M) = 0.
          19 CONTINUE
        18 CONTINUE
      17 CONTINUE
    16 CONTINUE
  15 CONTINUE
C
C GET INTERNAL B-FIELD PARAMETERS
  CALL GETIP(BFPAR,TPRM,BFLIM,BFEXST)
C
C SET UP WEIGHTING PARAMETERS FOR INTERPOLATION
C
  DO 10 II=1,5
    T(2,II)=(BFPAR(II)-BFLIM(1,II))/(BFLIM(2,II)-BFLIM(1,II))
    T(1,II)=1.-T(2,II)
  10 CONTINUE
C
C FIND MINIMUM CROSSING POINT FOR EACH OF THE "CORNERS"
DO 31 I = 1,2
  DO 32 J = 1,2
    DO 33 K = 1,2
      DO 34 L = 1,2
        DO 35 M = 1,2
          IF (.NOT.BFEXST(I,J,K,L,M)) THEN
            DO NOT ATTEMPT A FIELD-LINE TRACE
            GOTO 36
          END IF
          SET INTERNAL PARAMETERS
          STAND = TPRM( 1,I,J,K,L,M)
          TILT = TPRM( 2,I,J,K,L,M)
          HJNEAR = TPRM( 3,I,J,K,L,M)
          XNEAR = TPRM( 4,I,J,K,L,M)
          HJFRAC = TPRM( 5,I,J,K,L,M)
          DY = TPRM( 6,I,J,K,L,M)
          D = TPRM( 7,I,J,K,L,M)
          COLLAP = TPRM( 8,I,J,K,L,M)
          DELXC = TPRM( 9,I,J,K,L,M)
          DYC = TPRM(10,I,J,K,L,M)
          BRN = TPRM(11,I,J,K,L,M)
          BRP = TPRM(12,I,J,K,L,M)
          RN = TPRM(13,I,J,K,L,M)
          RP = TPRM(14,I,J,K,L,M)
        35 CONTINUE
      34 CONTINUE
    33 CONTINUE
  32 CONTINUE
31 CONTINUE
C
C SET-UP BFIELD INTERNAL COEFFICIENTS AND TAIL CURRENTS
ISWIT = 0
```

	CALL BFIELD (ISWIT,X,Y,Z,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,	MAP05080
1	COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BX,BY,BZ)	MAP05090
	CALL BFIELD (ISWIT,X,Y,Z,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,	MAP05100
1	COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BX,BY,BZ)	MAP05110
C		MAP05120
C	SINGLE MAGNETIC FIELD-LINE TRACE TO MINIMUM B SURFACE	MAP05130
	CALL BMNMAP (ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D, COLLAP,	MAP05140
1	DELXC,DYC,BRN,BRP,RN,RP,RADIUS,XSTRT,YSTRT,ZSTRT,	MAP05150
2	IOPEN,XHIT(I,J,K,L,M),YHIT(I,J,K,L,M),ZHIT(I,J,K,L,M))	MAP05160
C		MAP05170
	IF (IOPEN.EQ.1) THEN	MAP05180
	WRITE(6,*) 'OPEN FIELD-LINE FOUND, INTERPOLATION INVALIDATED'	MAP05190
	WRITE(6,*) 'EXITED FROM SUBROUTINE MAPTRACE'	MAP05200
	RETURN	MAP05210
	END IF	MAP05220
36	CONTINUE	MAP05230
35	CONTINUE	MAP05240
34	CONTINUE	MAP05250
33	CONTINUE	MAP05260
32	CONTINUE	MAP05270
31	CONTINUE	MAP05280
C		MAP05290
C	INITIALIZE RANGE SPREAD INDICATORS	MAP05300
	XMAX = -999.	MAP05310
	YMAX = -999.	MAP05320
	ZMAX = -999.	MAP05330
	XMIN = 999.	MAP05340
	YMIN = 999.	MAP05350
	ZMIN = 999.	MAP05360
C	MAIN INTERPOLATION LOOPS	MAP05370
	DO 21 N1=1,2	MAP05380
	N(1)=N1	MAP05390
	DO 22 N2=1,2	MAP05400
	N(2)=N2	MAP05410
	DO 23 N3=1,2	MAP05420
	N(3)=N3	MAP05430
	DO 24 N4=1,2	MAP05440
	N(4)=N4	MAP05450
	DO 25 N5=1,2	MAP05460
	N(5)=N5	MAP05470
C		MAP05480
	COEFF=1.	MAP05490
	DO 30 M=1,5	MAP05500
	COEFF=COEFF*T(N(M),M)	MAP05510
30	CONTINUE	MAP05520
C		MAP05530
	XMAP = XMAP + COEFF*XHIT(N1,N2,N3,N4,N5)	MAP05540
	YMAP = YMAP + COEFF*YHIT(N1,N2,N3,N4,N5)	MAP05550
	ZMAP = ZMAP + COEFF*ZHIT(N1,N2,N3,N4,N5)	MAP05560
C	UPDATE MAX-MIN RANGES	MAP05570
	IF (BFEXST(N1,N2,N3,N4,N5)) THEN	MAP05580
	IF (XHIT(N1,N2,N3,N4,N5).GT.XMAX) THEN	MAP05590
	XMAX = XHIT(N1,N2,N3,N4,N5)	MAP05600
	ELSE	MAP05610
	IF (XHIT(N1,N2,N3,N4,N5).LT.XMIN) THEN	MAP05620
	XMIN = XHIT(N1,N2,N3,N4,N5)	MAP05630
	END IF	MAP05640
	END IF	MAP05650
C		MAP05660
	IF (YHIT(N1,N2,N3,N4,N5).GT.YMAX) THEN	MAP05670
	YMAX = YHIT(N1,N2,N3,N4,N5)	MAP05680
	ELSE	MAP05690
	IF (YHIT(N1,N2,N3,N4,N5).LT.YMIN) THEN	MAP05700
	YMIN = YHIT(N1,N2,N3,N4,N5)	MAP05710

```
C
      END IF
      END IF
      IF (ZHIT(N1,N2,N3,N4,N5).GT.ZMAX) THEN
        ZMAX = ZHIT(N1,N2,N3,N4,N5)
      ELSE
        IF (ZHIT(N1,N2,N3,N4,N5).LT.ZMIN) THEN
          ZMIN = ZHIT(N1,N2,N3,N4,N5)
        END IF
      END IF
      END IF
      CONTINUE
25      CONTINUE
24      CONTINUE
23      CONTINUE
22      CONTINUE
21      CONTINUE
      X RANGE = XMAX - XMIN
      Y RANGE = YMAX - YMIN
      Z RANGE = ZMAX - ZMIN
      WRITE(6,*) 'MAPPING POINTS USED FOR INTERPOLATION WERE SPREAD:'
      WRITE(6,*) X RANGE,'RE ALONG THE XGSM DIRECTION'
      WRITE(6,*) Y RANGE,'RE ALONG THE YGSM DIRECTION'
      WRITE(6,*) Z RANGE,'RE ALONG THE ZGSM DIRECTION'
C
      RETURN
      END
C ***** GETIP *****
      SUBROUTINE GETIP(BFPR,TPRM,BFLIM,BFEXST)
C
C   PURPOSE-   FIND THE MSM PARAMETER SETS WHICH LIE CLOSEST TO THE
C               INPUT PARAMETER SET AND LOAD THE B-FIELD MODEL
C               INTERNAL PARAMETERS FROM THE APPROPRIATE MATRICES TO
C               BE PASSED ON FOR MAGNETIC FIELD LINE TRACING PURPOSES
C   INPUT-     BFPR: MSM INPUTS
C   OUTPUT-    TPRM: BFIELD MODEL INTERNAL INPUTS (14 FOR EACH)
C               BFLIM: VALUES FROM PARAMETER "GRID" WHICH BRACKET
C                       THE VALUES OF BFPR
C               BFEXST: INDICATES IF A BFIELD MODEL AS REPRESENTED
C                       BY A MATRIX HAS BEEN LOADED.
C   VARIABLES- BFNDX: INDEX ASSIGNED TO BFLIM VALUES
C               STNDPR,TILTPR,FINEDP,DSTPR,STCHPR: GRID VALUES OF THE
C               MSM INPUTS
C               MSTND,MTILT,MINED,MDST,MSTCH: NUMBER OF GRID POINTS
C               FOR EACH MSM INPUT
C               ICONF: NUMBER OF PARAMETER SETS LOADED.
C   SUBROUTINES-FNDBRK: FINDS THE SET OF B-FIELD MODELS WHICH
C                       BRACKET THE POINT OF INTEREST IN PARAMETER
C                       SPACE.
C               LOADIP: LOADS INTERNAL B-FIELD PARAMETERS FROM MATRIX
C                       FILES INTO VARIABLE TPRM.
C   FUNCTIONS-MEXIST: INDICATES IF A BFIELD MODEL EXISTS
C
      PARAMETER (IMSTND=5,IMTILT=5,IMINED=16,IMDST=8,IMSTCH=2)
C
      LOGICAL MEXIST
      INTEGER BFNDX(2,5)
      LOGICAL*1 BFEXST(2,2,2,2,2)
C
      DIMENSION STNDPR(IMSTND),TILTPR(IMTILT),FINEDP(IMINED)
      DIMENSION DSTPR(IMDST),STCHPR(IMSTCH)
      DIMENSION BFLIM(2,5),BFPR(5)
      DIMENSION TPRM(14,2,2,2,2,2)
      DATA MSTND/IMSTND/
      DATA MTILT/IMTILT/
```

MAP05720
MAP05730
MAP05740
MAP05750
MAP05760
MAP05770
MAP05780
MAP05790
MAP05800
MAP05810
MAP05820
MAP05830
MAP05840
MAP05850
MAP05860
MAP05870
MAP05880
MAP05890
MAP05900
MAP05910
MAP05920
MAP05930
MAP05940
MAP05950
MAP05960
MAP05970
MAP05980
MAP05990
MAP06000
MAP06010
MAP06020
MAP06030
MAP06040
MAP06050
MAP06060
MAP06070
MAP06080
MAP06090
MAP06100
MAP06120
MAP06130
MAP06140
MAP06150
MAP06160
MAP06170
MAP06200
MAP06210
MAP06220
MAP06230
MAP06240
MAP06110
MAP06250
MAP06260
MAP06270
MAP06290
MAP06300
MAP06310
MAP06320
MAP06330
MAP06340
MAP06350
MAP06360
MAP06370

90/06/30
12:04:14

mapint.for

11

```
DATA MINED/IMINED/
DATA MDST/IMDST/
DATA MSTCH/IMSTCH/
C
DATA STNDPR/6.0,8.0,10.0,12.0,14.0/
DATA TILTPR/-35.0,-17.5,0.0,17.5,35.0/
DATA FINEDP/49.47, 53.20, 55.89, 57.96, 59.60,
1      60.95, 62.06, 63.02, 63.99, 65.01,
2      65.91, 66.72, 67.45, 68.12, 68.73,
3      69.30/
DATA DSTPR/-400.0, -300.0, -200.0, -150.0,
1      -100.0, -50.0, 0.0, 50.0/
DATA STCHPR/0.0, 1.0/
C
EXTERNAL MEXIST
C
CALL FNDBRK(BFPAR(1),STNDPR,MSTND,BFNDX(1,1),BFNDX(2,1))
CALL FNDBRK(BFPAR(2),TILTPR,MTILT,BFNDX(1,2),BFNDX(2,2))
CALL FNDBRK(BFPAR(3),FINEDP,MINED,BFNDX(1,3),BFNDX(2,3))
CALL FNDBRK(BFPAR(4),DSTPR, MDST, BFNDX(1,4),BFNDX(2,4))
CALL FNDBRK(BFPAR(5),STCHPR,MSTCH,BFNDX(1,5),BFNDX(2,5))
C
DO 5, I=1,2
    BFLIM(I,1) = STNDPR(BFNDX(I,1))
    BFLIM(I,2) = TILTPR(BFNDX(I,2))
    BFLIM(I,3) = FINEDP(BFNDX(I,3))
    BFLIM(I,4) = DSTPR(BFNDX(I,4))
    BFLIM(I,5) = STCHPR(BFNDX(I,5))
5 CONTINUE
C
ICOUNT = 0
DO 10, I=1,2
    DO 20, J=1,2
        DO 30, K=1,2
            DO 40, L=1,2
                DO 50, M=1,2
                    IF (MEXIST(BFNDX(I,1),BFNDX(J,2),BFNDX(K,3),
1                      BFNDX(L,4),BFNDX(M,5))) THEN
                        CALL LOADIP(I,J,K,L,M,BFNDX,TPRM)
                        BFEXST(I,J,K,L,M) = .TRUE.
                        ICOUNT = ICOUNT + 1
                    ELSE
                        BFEXST(I,J,K,L,M) = .FALSE.
                    ENDIF
20 CONTINUE
30 CONTINUE
40 CONTINUE
50 CONTINUE
10 CONTINUE
    IF (ICOUNT.LT.8) THEN
        WRITE(6,1000) ICOUNT
1000  FORMAT(1X,'ONLY',I3,' MODELS AVAILABLE (NEED 8 MINIMUM)')
        WRITE(6,*) 'B-FIELD TRACING INTERPOLATION PROCEDURE INVALIDATED'
        WRITE(6,*) 'EXITED FROM SUBROUTINE GETIP'
        RETURN
    END IF
C
RETURN
END
C
***** LOADIP *****
C SUBROUTINE LOADIP(I,J,K,L,M,BFNDX,TPRM)
C
C PURPOSE-
```

MAP06380
MAP06390
MAP06400
MAP06410
MAP06420
MAP06430
MAP06440
MAP06450
MAP06460
MAP06470
MAP06480
MAP06490
MAP06500

MAP06510
MAP06540
MAP06550
MAP06560
MAP06570
MAP06580
MAP06590
MAP06600
MAP06610
MAP06620
MAP06630
MAP06640
MAP06650
MAP06660
MAP06670
MAP06680
MAP06690
MAP06700
MAP06710
MAP06720
MAP06730
MAP06740
MAP06750
MAP06760
MAP06770
MAP06780
MAP06790
MAP06800
MAP06810
MAP06820
MAP06830
MAP06840
MAP06850
MAP06860
MAP06870
MAP06880
MAP06890
MAP06900
MAP06910
MAP06920
MAP06930
MAP06940
MAP06950
MAP06960
MAP06970
MAP06980
MAP06990
MAP07000
MAP07010

90/06/30
12:04:14

12

mapint.for

```
C      THIS SUBROUTINE LOADS INTERNAL MODEL PARAMETERS FROM OFFLINE      MAP07020
C      B-SUPERMATRIX INTO THE TPRM MATRIX.  A CHECK IS MADE TO      MAP07030
C      VERIFY THAT THE CORRECT PARAMETERS HAVE BEEN RETRIEVED.  IF THIS      MAP07040
C      CHECK FAILS THE PROGRAM WILL STOP.  THIS SUBROUTINE HAS THE SAME      MAP07050
C      BASIC STRUCTURE (AND READS FROM THE SAME DATA FILES) AS SUBROUTINE      MAP07060
C      LOADBM.  ONE CALL PER MODEL.      MAP07070
C      INPUT-      I,J,K,L,M: INDICES FOR MATRIX BFNDX      MAP07080
C      BFNDX: INDEX NUMBERS OF PARAMETERS USED TO IDENTIFY      MAP07090
C      INDIVIDUAL MATRICES.      MAP07100
C      OUTPUT-      TPRM: INTERNAL B-FIELD PARAMETERS (14) FOR MAXIMUM OF 32      MAP07110
C      DIFFERENT MODELS (2X2X2X2X2)      MAP07120
C      VARIABLES-IDF1-IDF6: MATRIX IDENTIFIERS      MAP07130
C      MAP07140
C      INTEGER BFNDX(2,5)      MAP07150
C      CHARACTER*32 FILNAM      MAP07160
C      MAP07170
C      DIMENSION TPRM(14,2,2,2,2,2)      MAP07180
C      MAP07190
C      WRITE (FILNAM,1000) BFNDX(J,2), BFNDX(I,1), BFNDX(L,4),      MAP07200
1      BFNDX(K,3), (3-BFNDX(M,5))      MAP07210
1000  FORMAT ('BFIELD:BO',I1,I1,I1,I2.2,I1,'.DAT')      MAP07220
C      MAP07230
C      OPEN (UNIT=99,FILE=FILNAM,ERR=199,FORM='FORMATTED',      MAP07240
1      ACCESS='SEQUENTIAL')      MAP07250
C      MAP07260
C      READ (99,800) IDF1,IDF2,IDF3,IDF4,IDF5,IDF6,TPRM( 1,I,J,K,L,M),      MAP07270
1      TPRM( 2,I,J,K,L,M),TPRM( 3,I,J,K,L,M),TPRM( 4,I,J,K,L,M),      MAP07280
2      TPRM( 5,I,J,K,L,M),TPRM( 6,I,J,K,L,M),TPRM( 7,I,J,K,L,M),      MAP07290
3      TPRM( 8,I,J,K,L,M),TPRM( 9,I,J,K,L,M),TPRM(10,I,J,K,L,M),      MAP07300
4      TPRM(11,I,J,K,L,M),TPRM(12,I,J,K,L,M),TPRM(13,I,J,K,L,M),      MAP07310
5      TPRM(14,I,J,K,L,M)      MAP07320
800  FORMAT(3I1,I2.2,I1,I3,14F8.2)      MAP07330
C      MAP07340
C      IF ((BFNDX(I,1).NE.IDF2).OR.(BFNDX(J,2).NE.IDF1).OR.      MAP07350
1      (BFNDX(K,3).NE.IDF4).OR.(BFNDX(L,4).NE.IDF3).OR.      MAP07360
2      ((3-BFNDX(M,5)).NE.IDF5).OR.(IDF6.NE.201)) THEN      MAP07370
C      WRITE (6,*) 'FILE ',FILNAM,' IS INCORRECT.'      MAP07380
C      STOP 'INCORRECT B-FIELD. STOPPING IN LOADIP.'      MAP07390
C      ENDIF      MAP07400
C      MAP07410
C      CLOSE (99)      MAP07420
C      RETURN      MAP07430
C      MAP07440
199  CONTINUE      MAP07450
C      WRITE(6,*) 'STOPPING IN LOADIP. OPEN FAILED ON ',FILNAM      MAP07460
C      STOP 'STOPPING IN LOADBM. OPEN FAILED ON B-MATRIX.'      MAP07470
C      END      MAP07480
C***** BMNMAP      MAP07490
C      SUBROUTINE BMNMAP(ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,      MAP07500
1      COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,X,Y,Z,      MAP07510
2      IOPEN,XHIT,YHIT,ZHIT)      MAP07520
C      MAP07530
C      PURPOSE-      MAGNETIC FIELD LINE MAPPING TO THE MINIMUM |B| SURFACEMAP07540
C      OR THE MAGNETOPAUSE BOUNDARY, OPEN/CLOSED DETERMINED      MAP07550
C      INPUT-      FIRST 15: BFIELD INPUT      MAP07560
C      RADIUS: MAGNETOTAIL RADIUS FOR DEFINING BOUNDARY      MAP07570
C      X,Y,Z,: GSM STARTING POINT OF TRACE      MAP07580
C      OUTPUT-      IOPEN: DESIGNATES OPEN FIELD LINE      = 1      MAP07590
C      CLOSED FIELD LINE      = 0      MAP07600
C      XHIT,YHIT,ZHIT: LAST TRACE POINT. EITHER AT MSM      MAP07610
C      SURFACE WHEN FIELD LINE IS CLOSED OR AT A POINT      MAP07620
C      ON THE DEFINED MAGNETOPAUSE FOR OPEN LINES.      MAP07630
C      VARIABLE-      ER,EP,H:ERROR,DIRECTION, AND INITIAL TRACE STEP SIZE      MAP07640
C      DX1,DY1,DZ1,BB: VECTOR INFORMATION FOR SUBROUT. TRAC      MAP07650
```

mapint.for

```
C      BDIST: = RADIUS - STAND-OFF DISTANCE      MAP07660
C      DIST: DISTANCE FROM X-AXIS      MAP07670
C      XP: = XN + BDIST (DISTANCE FROM CYLINDER-SPHERE      MAP07680
C              INTERFACE IN X DIRECTION)      MAP07690
C      DDAY: DISTANCE FROM CYLINDER-SPHERE INTERFACE POINT      MAP07700
C              WHERE YGSM = ZGSM = 0      MAP07710
C      XA,YA,ZA: STARTING POINT FOR TRACING STEP      MAP07720
C      XN,YN,ZN: FINISH POINT FOR TRACING STEP      MAP07730
C      BMAGA,BMAGN: MAGNETIC FIELD MAGNITUDE AT POINTS      MAP07740
C              (XA,YA,ZA) AND (XN,YN,ZN) RESPECTIVELY.      MAP07750
C              USED WHEN TRACE DIRECTION IS BEING DECIDED.      MAP07760
C      BX,BY,BZ: MAGNETIC FIELD STRENGTH (NT) DURING TRACE.      MAP07770
C      BSIZE: MAGNETIC FIELD MAGNITUDE      MAP07780
C      BMIN: CURRENT SMALLEST B-FIELD MAGNITUDE      MAP07790
C      SUBROUTINE- TRAC: FINDS SEQUENTIAL STEPS IN VECTOR TRACE      MAP07800
C      BFIELD: MAGNETIC FIELD MODEL POINT CALCULATIONS      MAP07810
C      MAP07820
COMMON/BSTEP/DX1,DY1,DZ1,BB,H,EP,ER      MAP07830
C      MAP07840
C      SET TRACING ERROR LIMITS, INITIAL STEP SIZE, AND DIRECTION      MAP07850
ER = 0.25E-02      MAP07860
H = 0.25      MAP07870
EP = -1.0      MAP07880
C      INITIALIZE QUANTITIES      MAP07890
IOPEN = 1      MAP07900
XHIT = 999.      MAP07910
YHIT = 999.      MAP07920
ZHIT = 999.      MAP07930
XA = X      MAP07940
YA = Y      MAP07950
ZA = Z      MAP07960
DX1=0.      MAP07970
DY1=0.      MAP07980
DZ1=0.      MAP07990
BDIST = RADIUS - STAND      MAP08000
C      SET BMIN START VALUE, ASSUME START POINT |B| VALUE      MAP08010
CALL BFIELD(ISWIT,XA,YA,ZA,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,      MAP08020
1      COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BX,BY,BZ)      MAP08030
BMIN = SQRT(BX**2 + BY**2 + BZ**2)      MAP08040
C      CHECK THAT DIRECTION OF TRACE IS TOWARDS MINIMUM      MAP08050
CALL TRAC(ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,COLLAP,      MAP08060
1      DELXC,DYC,BRN,BRP,RN,RP,RADIUS,XA,YA,ZA,XN,YN,ZN)      MAP08070
C      GET FIELD STRENGTH AT STARTING POINT      MAP08080
CALL BFIELD(ISWIT,XA,YA,ZA,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,      MAP08090
1      COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BXA,BYA,BZA)      MAP08100
C      GET FIELD STRENGTH AT FIRST TRACER STEP POINT      MAP08110
CALL BFIELD(ISWIT,XN,YN,ZN,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,      MAP08120
1      COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BXN,BYN,BZN)      MAP08130
BMAGA = SQRT(BXA**2 + BYA**2 + BZA**2)      MAP08140
BMAGN = SQRT(BXN**2 + BYN**2 + BZN**2)      MAP08150
IF (BMAGN.GT.BMAGA) THEN      MAP08160
C      MOVING AWAY FROM MINIMUM, CHANGE DIRECTION      MAP08170
EP = -EP      MAP08180
CALL TRAC(ISWIT,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,COLLAP,      MAP08190
1      DELXC,DYC,BRN,BRP,RN,RP,RADIUS,XA,YA,ZA,XN,YN,ZN)      MAP08200
C      GET FIELD STRENGTH AT FIRST TRACER STEP POINT IN NEW DIR.      MAP08210
CALL BFIELD(ISWIT,XN,YN,ZN,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,      MAP08220
1      COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BXN,BYN,BZN)      MAP08230
BMAGN = SQRT(BXN**2 + BYN**2 + BZN**2)      MAP08240
IF (BMAGN.GT.BMAGA) THEN      MAP08250
C      FIRST POINT IN EITHER DIRECTION HAS STRONGER FIELD      MAP08260
C      MUST HAVE STARTED ON THE MINIMUM |B| SURFACE      MAP08270
IOPEN = 0      MAP08280
XHIT = XA      MAP08290
```

90/06/30
12:04:14

mapint.for

14

```

      YHIT = YA
      ZHIT = ZA
      GOTO 20
    END IF
  END IF
20  CONTINUE
C    START FULL TRACE OF A SINGLE FIELD-LINE
  CALL TRAC (ISWIT, STAND, TILT, HJNEAR, XNEAR, HJFRAC, DY, D, COLLAP,
1    DELXC, DYC, BRN, BRP, RN, RP, RADIUS, XA, YA, ZA, XN, YN, ZN)
  DIST = SQRT (YN**2 + ZN**2)
  XP = XN + BDIST
  DDAY = SQRT (XP**2 + YN**2 + ZN**2)
  IF ((XN.LT. (-50.)).OR. (DIST.GT.RADIUS)).OR.
1    ((XP.GE.0.).AND. (DDAY.GT.RADIUS))) THEN
C    TRACE HIT MAGNETOPAUSE OR FAR TAIL LIMIT (OPEN FIELD LINE)
    IOPEN = 1
    XHIT = XN
    YHIT = YN
    ZHIT = ZN
    WRITE (6,*) '*****'
    WRITE (6,*) 'OPEN FIELD-LINE HIT AT POINT: XGSM    YGSM    ZGSM'
    WRITE (6,*) XHIT, YHIT, ZHIT
    GOTO 20
  END IF
C
  CALL BFIELD (ISWIT, XN, YN, ZN, STAND, TILT, HJNEAR, XNEAR, HJFRAC, DY, D,
1  COLLAP, DELXC, DYC, BRN, BRP, RN, RP, RADIUS, BX, BY, BZ)
  BSIZE = SQRT (BX**2 + BY**2 + BZ**2)
  IF (BSIZE.LE.BMIN) THEN
C    CLOSER TO MINIMUM IN |B|
    BMIN = BSIZE
    XHIT = XN
    YHIT = YN
    ZHIT = ZN
    IOPEN = 0
  ELSE
C    PAST MINIMUM |B|, DONE
    GOTO 20
  END IF
C
  XA = XN
  YA = YN
  ZA = ZN
  GOTO 30
C
20  CONTINUE
  RETURN
  END
C***** TRAC
  SUBROUTINE TRAC (ISWIT, STAND, TILT, HJNEAR, XNEAR, HJFRAC, DY, D, COLLAP,
1  DELXC, DYC, BRN, BRP, RN, RP, RADIUS, XA, YA, ZA, XN, YN, ZN)
C
C  PURPOSE-    FINDS SUCCESSIVE STEP LOCATION IN A TRACE OF A VECTOR
C              FIELD, E.G., USED TO MAP MAGNETIC FIELD LINES.
C  INPUT-     FIRST 15 ARE BFIELD MODEL INPUTS (SUBROUTINE BFIELD)
C              XA,YA,ZA: GSM LOCATION OF START POINT OF VECTOR TRACE.
C  OUTPUT-    RADIUS: MAGNETOTAIL RADIUS (RE)
C              XN,YN,ZN: GSM LOCATION OF NEXT STEP IN VECTOR TRACE.
C  VARIABLES- H: STEP SIZE
C              EP: DIRECTION OF TRACE (+1. IS PARALLEL TO FIELD
C                  VECTOR AND -1. IS ANTIPARALLEL TO SAME)
C              ER: ALLOWABLE DISTANCE ERROR LIMIT (RE)
C  SUBROUTINES-FUNC: PROVIDES VECTOR INFORMATION ABOUT THE BFIELD

```

90/06/30
12:04:14

mapint.for

15

```
COMMON/BSTEP/DX1,DY1,DZ1,BB,H,EP,ER
REAL DX1,DY1,DZ1,BB,H,EP,ER,XA,YA,ZA,XN,YN,ZN
C
1 CALL FUNC(ISWIT,XA,YA,ZA,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
1 COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,DX1,DY1,DZ1,BF1)
C
1 CONTINUE
H1=H*EP
H2=0.5*H1
C
XN1=XA+H1*DX1
YN1=YA+H1*DY1
ZN1=ZA+H1*DZ1
XN2=XA+H2*DX1
YN2=YA+H2*DY1
ZN2=ZA+H2*DZ1
C
1 CALL FUNC(ISWIT,XN2,YN2,ZN2,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
1 COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,DX2,DY2,DZ2,BF2)
C
XN3=XN2+H2*DX2
YN3=YN2+H2*DY2
ZN3=ZN2+H2*DZ2
C
DELX=XN1-XN3
DELY=YN1-YN3
DELZ=ZN1-ZN3
RTES1=DELX*DELX
RTES2=DELY*DELY
RTES3=DELZ*DELZ
RTES=SQRT(RTES1+RTES2+RTES3)
C
4 IF(RTES-ER) 4,4,2
4 CONTINUE
C
XN=XN3-DELX
YN=YN3-DELY
ZN=ZN3-DELZ
BB=BF2
7 IF(RTES-(0.25*ER)) 7,6,6
7 H=2.*H
C ***LIMIT MAXIMUM STEP SIZE***
IF (H.GT.(5.0)) THEN
H = 5.0
END IF
6 GOTO 5
C
2 CONTINUE
H=0.5*H
GOTO 1
C
5 CONTINUE
RETURN
END
C***** FUNC
SUBROUTINE FUNC(ISWIT,X,Y,Z,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,
1 COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,DX,DYY,DZ,B)
C
C PURPOSE- PROVIDE VECTOR INFORMATION ABOUT A FUNCTION (BFIELD)
C INPUT- FIRST 18: BFIELD INPUTS
C OUTPUT- DX,DYY,DZ,B: VECTOR COMPONENT INFORMATION
C SUBROUTINE-BFIELD: MAGNETIC FIELD MODEL
C
BX = 0.
BY = 0.
BZ = 0.
```

MAP08940
MAP08950
MAP08960
MAP08970
MAP08980
MAP08990
MAP09000
MAP09010
MAP09020
MAP09030
MAP09040
MAP09050
MAP09060
MAP09070
MAP09080
MAP09090
MAP09100
MAP09110
MAP09120
MAP09130
MAP09140
MAP09150
MAP09160
MAP09170
MAP09180
MAP09190
MAP09200
MAP09210
MAP09220
MAP09230
MAP09240
MAP09250
MAP09260
MAP09270
MAP09280
MAP09290
MAP09300
MAP09310
MAP09320
MAP09330
MAP09340
MAP09350
MAP09360
MAP09370
MAP09380
MAP09390
MAP09400
MAP09410
MAP09420
MAP09430
MAP09440
MAP09450
MAP09460
MAP09470
MAP09480
MAP09490
MAP09500
MAP09510
MAP09520
MAP09530
MAP09540
MAP09550
MAP09560
MAP09570

90/06/30
12:04:14

16

mapint.for

```
CALL BFIELD (ISWIT,X,Y,Z,STAND,TILT,HJNEAR,XNEAR,HJFRAC,DY,D,  
1 COLLAP,DELXC,DYC,BRN,BRP,RN,RP,RADIUS,BX,BY,BZ)  
C  
B = SQRT(BX**2 + BY**2 + BZ**2)  
DX = BX/B  
DYY = BY/B  
DZ = BZ/B  
C  
RETURN  
END  
SUBROUTINE CHECK(IYEAR,IDAY,IHOUR,MINUTE,ISCNDS,IFLAG)  
C THIS SUBROUTINE CHECKS THAT THE INTERACTIVE INPUT IS WITHIN  
C REASONABLE LIMITS.  
C  
IFLAG=0  
IF((IYEAR.LT.0).OR.(IYEAR.GT.99))THEN  
WRITE(6,*)' INCORRECT INPUT FOR YEAR-MUST BE BETWEEN 0 AND 99'  
IFLAG= 1  
RETURN  
ENDIF  
IF((IDAY.LT.1).OR.(IDAY.GT.366))THEN  
WRITE(6,*)' INCORRECT INPUT FOR DAY-MUST BE BETWEEN 1 AND 366'  
IFLAG= 1  
RETURN  
ENDIF  
IF((IHOUR.LT.0).OR.(IHOUR.GT.24))THEN  
WRITE(6,*)' INCORRECT INPUT FOR HOUR-MUST BE BETWEEN 1 AND 24'  
IFLAG=1  
RETURN  
ENDIF  
IF((MINUTE.LT.0).OR.(MINUTE.GT.60))THEN  
WRITE(6,*)' INCORRECT INPUT FOR MINUTE-MUST BE BETWEEN 0 AND 60'  
IFLAG= 1  
RETURN  
ENDIF  
IF((ISCNDS.LT.0).OR.(ISCNDS.GT.60))THEN  
WRITE(6,*)' INCORRECT INPUT FOR SECONDS-MUST BE BETWEEN 0 AND 60'  
IFLAG= 1  
RETURN  
ENDIF  
RETURN  
END  
SUBROUTINE TCHECK(IYEAR,IDAY,IHOUR,MINUTE,ISCNDS,AUGPAR,AGPAR1,  
1 NAUGEL,IFLAG,ISTART)  
C  
C THIS SUBROUTINE CHECKS TO FIND THE RECORD NUMBERS OF THE DATA  
C FILES WHICH SPAN THE REQUESTED TIME.  
C  
DIMENSION AUGPAR(NAUGEL),AGPAR1(NAUGEL),ISTART(3),ITIME(3)  
IFLAG=0  
ITIME(1)=IYEAR  
ITIME(2)=IDAY  
ITIME(3)=IHOUR*3600 + MINUTE*60 + ISCNDS  
TIMCHK=TCONV3(ITIME,ISTART)  
ITIME(1)=AUGPAR(1)  
ITIME(2)=AUGPAR(2)  
ITIME(3)=AUGPAR(3)*3600 + AUGPAR(4)*60 + AUGPAR(5)  
TIMAUG=TCONV3(ITIME,ISTART)  
ITIME(1)=AGPAR1(1)  
ITIME(2)=AGPAR1(2)  
ITIME(3)=AGPAR1(3)*3600 + AGPAR1(4)*60 + AGPAR1(5)  
TIMAG1=TCONV3(ITIME,ISTART)  
IF((TIMCHK.GE.TIMAUG).AND.(TIMCHK.LT.TIMAG1))THEN
```

MAP09580
MAP09590
MAP09600
MAP09610
MAP09620
MAP09630
MAP09640
MAP09650
MAP09660
MAP09670
MAP09680
MAP09690
MAP09700
MAP09710
MAP09720
MAP09730
MAP09740
MAP09750
MAP09760
MAP09770
MAP09780
MAP09790
MAP09800
MAP09810
MAP09820
MAP09830
MAP09840
MAP09850
MAP09860
MAP09870
MAP09880
MAP09890
MAP09900
MAP09910
MAP09920
MAP09930
MAP09940
MAP09950
MAP09960
MAP09970
MAP09980
MAP09990
MAP10000
MAP10010
MAP10020
MAP10030
MAP10040
MAP10050
MAP10060
MAP10070
MAP10080
MAP10090
MAP10100
MAP10110
MAP10120
MAP10130
MAP10140
MAP10150
MAP10160
MAP10170
MAP10180
MAP10190
MAP10200
MAP10210

```

C      IFLAG=0
C      ELSE
C      IFLAG=1
C      ENDIF
C      RETURN
C      END
C      ***** TCONV3 *****
C      REAL FUNCTION TCONV3(ITIME,ISTART)
C      THIS FUNCTION TAKES A START YEAR, START DAY, A CURRENT YEAR,
C      A CURRENT DAY, AND A CURRENT SECOND.. IT PRODUCES
C      A VALUE THAT IS THE SECONDS FROM THE START YEAR AND DAY.
C      THIS VALUE IS RELATIVE TO MIDNIGHT OF THE START DAY.
C      PARAMETER (IYR=1,IDY=2,ISEC=3)
C      DIMENSION ITIME(3),ISTART(3)
C      INTRINSIC REAL
C      IF (ISTART(IYR).EQ.ITIME(IYR)) THEN
C      1      TEMP = REAL(ITIME(IDY) - ISTART(IDY))*86400.0 +
C      REAL(ITIME(ISEC))
C      ELSEIF (MOD(ISTART(IYR),4).EQ.0) THEN
C      1      TEMP = REAL(ITIME(IDY) + 366 - ISTART(IDY))*86400.0 +
C      REAL(ITIME(ISEC))
C      ELSE
C      1      TEMP = REAL(ITIME(IDY) + 365 - ISTART(IDY))*86400.0 +
C      REAL(ITIME(ISEC))
C      ENDIF
C      TCONV3 = TEMP
C      RETURN
C      END
C      SUBROUTINE LOCATE(RVAL,PVAL,LATDIM,LTDIM,R,P,BI,BJ)
C      PURPOSE:  FIND GRID LOCATION (BI,BJ) OF PHYSICAL LOCATIONS GIVEN BY
C      (RVAL,PVAL,LL)
C      VERSION 1.00
C      DATE: 01.22.90
C      INPUT:
C      RVAL      RADIAL DISTANCE (RE)
C      PVAL      HOUR ANGLE MEASURED EASTWARD FROM NOON (RADIAN)
C      LATDIM    NUMBER OF LATITUDINAL GRID PTS
C      LTDIM     MAX NUMBER OF LOCAL TIME GRID PTS (INCL WRAPAROUND)
C      R         RADIAL DISTANCE ARRAY (RE)
C      P         HOUR ANGLE ARRAY (RADIAN)
C      OUTPUT:
C      BI        NON-INTEG I LOCATION OF (RVAL,XLT)
C      BJ        NON-INTEG J LOCATION OF (RVAL,XLT)
C      PROGRAMMER: R.W. SPIRO
C      DIMENSION R(LATDIM,LTDIM),P(LATDIM,LTDIM)

```

90/06/30
12:04:14

mapint.for

18

```
PI=ATAN2(0.,-1.)
C
C
C
C
C
C
C
C
C SEARCH GRID AND TRY TO LOCALIZE (RVAL,PVAL) WITHIN A GRID SQUARE
  DO 30 J=2,LTDIM-2
    JJ=J
C
    DO 40 I=2,LATDIM
      II=I
C
C GET PVAL,P(I,J), AND P(I,J+1) IN SAME MODULUS
      CALL PFIX(PVAL,P(I,J),P(I,J+1),P4,P5)
C      WRITE(6,*) I,J,RVAL,R(I-1,J,LL),R(I,J,LL),PVAL,P4,P5
C
      IF(RVAL.LE.R(I-1,J)
2        .AND.
3          RVAL.GT.R(I,J)
4        .AND.
5          PVAL.GE.P4
6        .AND.
7          PVAL.LT.P5)
8        THEN
          GO TO 50
        END IF
C
40    CONTINUE
30  CONTINUE
C
  WRITE(6,*) 'STOPPING IN LOCATE'
  WRITE(6,*) 'UNABLE TO FIND (BI,BJ)'
  WRITE(6,*) 'RVAL= ',RVAL,' PVAL= ',PVAL
  STOP
C
C
50 CONTINUE
C
  DO 45 IK=2,LATDIM
    IIK=IK
    IF(RVAL.LE.R(IK-1,J+1).AND.RVAL.GT.R(IK,J+1)) GO TO 60
45 CONTINUE
C
  WRITE(6,*) 'STOPPING IN EFLOC'
  WRITE(6,*) 'UNABLE TO FIND (BI,BJ)'
  WRITE(6,*) 'RVAL= ',RVAL,' PVAL= ',PVAL
  STOP
C
60 CONTINUE
C
C COMPUTE COEFFICIENTS FOR INTERPOLATION
  F1=(RVAL-R(II-1,JJ))/(R(II,JJ)-R(II-1,JJ))+FLOAT(II-1)
  F2=(RVAL-R(IIK-1,JJ+1))/(R(IIK,JJ+1)-R(IIK-1,JJ+1))+
2    FLOAT(IIK-1)
C
  CALL PFIX(XLT,P(II,JJ),P(IIK,JJ+1),P4,P5)
C
  BJ=(PVAL-P4)/(P5-P4)+FLOAT(JJ)
C
  IF(BJ.LT.1.) BJ=BJ+FLOAT(LTDIM-3)
```

MAP10860
MAP10870
MAP10880
MAP10890
MAP10900
MAP10910
MAP10920
MAP10930
MAP10940
MAP10950
MAP10960
MAP10970
MAP10980
MAP10990
MAP11000
MAP11010
MAP11020
MAP11030
MAP11040
MAP11050
MAP11060
MAP11070
MAP11080
MAP11090
MAP11100
MAP11110
MAP11120
MAP11130
MAP11140
MAP11150
MAP11160
MAP11170
MAP11180
MAP11190
MAP11200
MAP11210
MAP11220
MAP11230
MAP11240
MAP11250
MAP11260
MAP11270
MAP11280
MAP11290
MAP11300
MAP11310
MAP11320
MAP11330
MAP11340
MAP11350
MAP11360
MAP11370
MAP11380
MAP11390
MAP11400
MAP11410
MAP11420
MAP11430
MAP11440
MAP11450
MAP11460
MAP11470
MAP11480
MAP11490

90/06/30
12:04:14

19

mapint.for

```
C      IF (BJ.GT.FLOAT(LTDIM)) BJ=BJ-FLOAT(LTDIM-3)
C
C      BI=F1+(BJ-FLOAT(JJ))*(F2-F1)/(FLOAT(JJ+1)-FLOAT(JJ))
C
C      RETURN
C      END
C
C      SUBROUTINE PFIX(P1,P2,P3,P4,P5)
C
C      PURPOSE:  SUBROUTINE TO ADJUST MODULUS OF P2 AND P3 TO MATCH P1.
C                ON OUTPUT P4 CORRESPONDS TO P2, P5 TO P3
C
C      VERSION 1.00                                DATE: 09.04.89
C
C      PROGRAMMER: R.W. SPIRO
C
C      PI=ATAN2(0.,-1.)
C
C      P4=P2
C      P5=P3
C
C      IF (ABS(P2-P3).GT.PI) THEN
C        IF (P2.LT.P3) THEN
C          P4=P2+2.*PI
C        ELSE
C          P5=P3+2.*PI
C        END IF
C
C        IF (P1.LT.PI) THEN
C          P4=P4-2.*PI
C          P5=P5-2.*PI
C        END IF
C
C      END IF
C
C      RETURN
C      END
C      FUNCTION G3NTRP(A,IMAX,JMAX,KMAX,BI,BJ,BK)
C
C      VERSION 1.00                                DATE: 01.11.88
C      1.01A                                         02.02.89
C
C      PURPOSE:  FUNCTION SUBPROGRAM TO PERFORM A GENERAL 3-D LINEAR
C                INTERPOLATION OF ARRAY A(I,J,K) AT PT(BV(1),BV(2),BV(3))
C
C      INPUT:
C        A                      3-D ARRAY TO BE INTERPOLATED
C                                INTERPOLATE A
C
C      COMMON /LUNIT/ LUERR, LUPPT, LUCORD, LUPRNT, LUIDAT, LUHDYE,
C 1      LSPARE, LUETAB, LUEBEG, LUPLEC, LUPION, LUEAVG, LUFLSM, LUETA,
C 2      LUEFLX, LUEFLD, LUBFLD, LUBMIN, LUXMIN, LUYMIN, LUZMIN, LUALOC,
C 3      LUCOLT, LUBNDL, LUPDAT, LUVN, LUVS, LIONEG, LIONNO, LUENCH
C      DIMENSION A(IMAX,JMAX,KMAX)
C      DIMENSION NDX(3),NDIM(3),BV(3),COEF(3,2)
C
C      PREPARE INDICES FOR INTERPOLATION
```

MAP11500
MAP11510
MAP11520
MAP11530
MAP11540
MAP11550
MAP11560
MAP11570
MAP11580
MAP11590
MAP11600
MAP11610
MAP11620
MAP11630
MAP11640
MAP11650
MAP11660
MAP11670
MAP11680
MAP11690
MAP11700
MAP11710
MAP11720
MAP11730
MAP11740
MAP11750
MAP11760
MAP11770
MAP11780
MAP11790
MAP11800
MAP11810
MAP11820
MAP11830
MAP11840
MAP11850
MAP11860
MAP11870
MAP11880
MAP11890
MAP11900
MAP11910
MAP11920
MAP11930
MAP11940
MAP11950
MAP11960
MAP11970
MAP11980
MAP11990
MAP12000
MAP12010
MAP12020
MAP12030
MAP12040
MAP12050
MAP12060
MAP12070
MAP12080
MAP12090
MAP12100
MAP12110
MAP12120
MAP12130

90/06/30
12:04:14

mapint.for

20

```
C
NDIM(1)=IMAX
NDIM(2)=JMAX
NDIM(3)=KMAX
BV(1)=BI
BV(2)=BJ
BV(3)=BK
DO 10 L=1,3
  NDX(L)=BV(L)
  IF (NDX(L).LT.1) NDX(L)=1
  IF (NDX(L).GT.NDIM(L)-1) NDX(L)=NDIM(L)-1
  IF (NDX(L).LE.0) NDX(L)=1
C
  FNDX=REAL(NDX(L))
  COEF(L,1)=1.-BV(L)+FNDX
  COEF(L,2)=BV(L)-FNDX
10 CONTINUE
C
C
G3NTRP=0.
KSTOP = MIN(KMAX,2)
JSTOP = MIN(JMAX,2)
DO 20 I=1,2
DO 20 J=1,JSTOP
DO 20 K=1,KSTOP
  G3NTRP=G3NTRP+
1      COEF(1,I)*COEF(2,J)*COEF(3,K)*A(NDX(1)+I-1,NDX(2)+J-1,
2      NDX(3)+K-1)
20 CONTINUE
RETURN
END
C
C
SUBROUTINE READ3D(LUN,FILNAM,LREC,IDIM,JDIM,KDIM,ITMDIM,ID,RID,
2      CHID,ARRAY)
C
PROGRAMMER: BOB SPIRO          DATE: 6/26/88
C
PURPOSE:  THIS SUBROUTINE READS A LOGICAL RECORD FROM THE
STANDARD MAGNETOSPHERIC SPECIFICATIONS MODEL FILE FORMAT.
C
CHARACTER CHID*80, FILNAM*6
C
  DIMENSION ARRAY(IDIM,JDIM,KDIM)
  DIMENSION ID(20),RID(20)
C
  OPEN THE FILE
C
  IRECL = IDIM*JDIM + 2*80
  OPEN (UNIT=LUN,ACCESS='DIRECT',RECL=IRECL,STATUS='OLD')
C
  READ THE ARRAY
C
  DO 10, K=1, KDIM
C
    CALCULATE THE RECORD NUMBER
C
    IREC = (LREC-1)*KDIM + K
    READ (LUN,REC=IREC) ID,RID,CHID,
2      ((ARRAY(I,J,K),I=1,IDIM),J=1,JDIM)
10 CONTINUE
C
CLOSE(LUN)
```

MAP12140
MAP12150
MAP12160
MAP12170
MAP12180
MAP12190
MAP12200
MAP12210
MAP12220
MAP12230
MAP12240
MAP12250
MAP12260
MAP12270
MAP12280
MAP12290
MAP12300
MAP12310
MAP12320
MAP12330
MAP12340
MAP12350
MAP12360
MAP12370
MAP12380
MAP12390
MAP12400
MAP12410
MAP12420
MAP12430
MAP12440
MAP12450
MAP12460
MAP12700
MAP12710
MAP12720
MAP12730
MAP12740
MAP12750
MAP12760
MAP12770
MAP12780
MAP12790
MAP12800
MAP12810
MAP12820
MAP12830
MAP12840
MAP12850
MAP12860

MAP12880
MAP12890
MAP12900
MAP12910
MAP12920
MAP12930
MAP12940
MAP12950
MAP12960
MAP12970
MAP12980
MAP12990
MAP13000

90/06/30
12:04:14

mapint.for

21

```
C
      RETURN
C
      END
C
***** FNDBRK *****
C
      SUBROUTINE FNDBRK(PARVAL,PVALS,IPDIM,MIN,MAX)
C
      THIS SUBROUTINE FINDS THE INDICES OF THE VALUES IN ARRAY
C      PVALS THAT BRACKET PARVAL.
C
      BRYAN BALES          9/12/89
C
      DIMENSION PVALS(IPDIM)
C
      MIN = 0
C
10    CONTINUE
      MIN = MIN + 1
      MAX = MIN + 1
C
      IF (PARVAL.LE.PVALS(MIN)) THEN
C
      MAX = MIN
C
      ELSEIF (PARVAL.GT.PVALS(MAX).AND.MAX.EQ.IPDIM) THEN
C
      MIN = MAX
C
      IF (PARVAL.GT.PVALS(MAX).AND.MAX.LT.IPDIM) THEN
        GO TO 10
      ENDIF
C
      RETURN
C
      END
C
      LOGICAL FUNCTION MEXIST(MSTND,MTILT,MINED,MDST,MSTCH)
C
      THIS FUNCTION CHECKS TO SEE IF A PARTICULAR B-FIELD IS
C      AVAILABLE.
C
      CHARACTER*32 FILNAM
      LOGICAL TEST
C
      WRITE (FILNAM,1000) MTILT,MSTND,MDST,MINED,(3-MSTCH)
1000  FORMAT ('BFIELD:BO',I1,I1,I1,I2.2,I1,'.DAT')
C
      INQUIRE (FILE=FILNAM,EXIST=TEST)
      MEXIST = TEST
C
      RETURN
C
      END
C
      SUBROUTINE RDVEC(NOUNIT,IRDIN,IEMAX,IEDIM,ALAM,THRSH,ERSHFT,IFLAV)
C
      THIS SUBROUTINE READS THE HEADER VECTOR FOR THE ETA FILES. THE
C      HEADER CONTAINS THE ALAM, THRESHOLD ENERGY, AND ERROR INFORMATION
C      FOR THIS RUN.
C
      DIMENSION ALAM(IEDIM),THRSH(IEDIM),ERSHFT(IEDIM),IFLAV(IEDIM)
C
      OPEN THE FILE
C
      IRECL = 4*iedim + 1
      OPEN(UNIT=NOUNIT,ACCESS='DIRECT',RECL=IRECL,STATUS='OLD')
C
      READ VALUES
C
      READ (NOUNIT,REC=IRDIN) IEMAX,ALAM,THRSH,ERSHFT,IFLAV
```

22

MAP14650
MAP14660
MAP14670
MAP14680
MAP14690
MAP14700
MAP14710
MAP14720
MAP14730
MAP14740
MAP14750
MAP14760
MAP14770
MAP14780
MAP14790
MAP14800
MAP14810
MAP14820
MAP14830
MAP14840
MAP14850
MAP14860
MAP14870
MAP14880
MAP14890
MAP14900
MAP14910
MAP14920
MAP14930
MAP14940
MAP14950
MAP14960
MAP14970
MAP14980
MAP14990
MAP15000
MAP15010
MAP15020
MAP15030
MAP15040
MAP15050
MAP15060
MAP15070
MAP15080
MAP15090
MAP15100
MAP15110
MAP15120
MAP15130
MAP15140
MAP15150
MAP15160
MAP15170
MAP15180
MAP15190
MAP15200
MAP15210
MAP15220
MAP15230
MAP15240
MAP15250
MAP15260
MAP15270
MAP15280

90/06/30
12:04:14

23

mapint.for

```
C      X,Y,Z: GSM COORDINATES IN EARTH RADII      MAP15290
C      STAND: STAND-OFF DISTANCE IN RE            MAP15300
C      TILT: DIPOLE TILT ANGLE IN DEGREES          MAP15310
C                                                    MAP15320
C      OUTPUT- BXTOT,BYTOT,BZTOT: MAGNETIC FIELD COMPONENTS IN GAMMA MAP15330
C      HJXS,HJYS,HJZS: CURRENT DENSITY COMPONENTS (AMPS/M**2) MAP15340
C      THE INPUT VARIABLES FOR SUBROUT. BVOIGT,TAILC, AND MAP15350
C      RINGC ARE EXPLAINED IN EACH RESPECTIVE SUBROUT. MAP15360
C                                                    MAP15370
C      VARIABLES-XNST,ZNST: XGSM AND ZGSM POSITIONS OF THE CURRENT SHEET MAP15380
C      IN THE NOON-MIDNIGHT MERIDIAN. SEGMENT MAP15390
C      BOUNDAFORM XNST(I),ZNST(I) MAP15400
C      BNST: CURRENT STRENGTH AT POINT XNST(I),ZNST(I) ABOVE MAP15410
C      RADIUS: MAGNETOPOUSE RADIUS IN RE MAP15420
C                                                    MAP15430
C      SUBROUTINES-GETRAD: CALCULATES RADIUS FROM STAND-OFF DISTANCE MAP15440
C      BVOIGT: FINDS VACUUM SHIELDED DIPOLE CONTRIBUTIONS MAP15450
C      ZPOS: CALCULATES COEFFICIENTS FOR PLACEMENT OF TAIL MAP15460
C      CURRENT SHEET MAP15470
C      SEGMENT: FINDS XNST,ZNST, AND BNST FOR NSTEP SEGMENTS MAP15480
C      (NOT CALLED IF HJNEAR = 0) MAP15490
C      TAILC: FINDS TAIL CURRENT SHEET CONTRIBUTIONS MAP15500
C      (NOT CALLED IF HJNEAR = 0.) MAP15510
C      RINGC: FINDS RING CURRENT SYSTEM CONTRIBUTIONS MAP15520
C      (NOT CALLED IF BRN = BRP = 0.) MAP15530
C      TCOLLP: FINDS COLLAPSING TAIL CURRENT CONTRIBUTIONS MAP15540
C      (NOT CALLED IF COLLAP = 1.) MAP15550
C      MAP15560
C      DIMENSION XNST(50),ZNST(50),BNST(50) MAP15570
C      MAP15580
C      FIRST CALL MUST HAVE ISWIT = 0 MAP15590
C      ISWIT = ISWIT + 1 MAP15600
C      *** LIMIT MAX VALUE OF ISWIT TO 4 *** MAP15610
C      IF (ISWIT.GT.3) THEN MAP15620
C      ISWIT = 4 MAP15630
C      END IF MAP15640
C      MAP15650
C      GET BVOIGT COEFFICIENTS MAP15660
C      IF (ISWIT.EQ.1) THEN MAP15670
C      MAP15680
C      MAP15690
C      SET CONSTANTS FOR VACUUM VOIGT SHIELDED DIPOLE MODEL MAP15700
C      NO PENETRATION OF IMF MAP15710
C      ALFA = 1.00 MAP15720
C      CDIPOL = 0.0 MAP15730
C      CIMF = 0.0 MAP15740
C      BIMF = 0.0 MAP15750
C      DIR = 0.0 MAP15760
C      MAP15770
C      ZERO QUANTITIES MAP15780
C      BXV = 0. MAP15790
C      BYV = 0. MAP15800
C      BZV = 0. MAP15810
C      BXT = 0. MAP15820
C      BYT = 0. MAP15830
C      BZT = 0. MAP15840
C      BXTC = 0. MAP15850
C      BYTC = 0. MAP15860
C      BZTC = 0. MAP15870
C      BXR = 0. MAP15880
C      BYR = 0. MAP15890
C      BZR = 0. MAP15900
C      MAP15910
C      GET RADIUS FROM STAND MAP15920
```

90/06/30
12:04:14

mapint.for

24

```
C      CALL GETRAD(STAND,RADIUS)
C      GET  COEFFICIENTS FOR BVOIGT
C      CALL BVOIGT(STAND,RADIUS,TILT,ALFA,CDIPOL,CIMF,BIMF,DIR,
1      ISWIT,X,Y,Z,GARB1,GARB2,GARB3)
C      END IF
C
C      SET CONSTANTS IN BVOIGT,SET-UP TAIL CURRENT WITH POSITION
C      IF (ISWIT.EQ.2) THEN
C      CALL BVOIGT(STAND,RADIUS,TILT,ALFA,CDIPOL,CIMF,BIMF,DIR,
1      ISWIT,X,Y,Z,GARB1,GARB2,GARB3)
C      CALCULATE COEFFICIENTS FOR CURRENT SHEET POSITION
C      CALL ZPOS(ISWIT,TILT,X,Z)
C      ADVANCE ISWIT BECAUSE ZPOS CALLED NEXT IN SEGMNT
C      ISWIT = ISWIT + 1
C      CALCULATE SEGMENT BOUNDARIES AND CURRENT STRENGTHS
C      IF (HJNEAR.NE.0.) THEN
C      CALL SEGMNT(ISWIT,TILT,XNEAR,HJNEAR,HJFRAC,NSTEP,
1      XNST,ZNST,BNST)
C      END IF
C      END IF
C
C      GET POINT VALUES
C      IF (ISWIT.EQ.4) THEN
C      CALL BVOIGT(STAND,RADIUS,TILT,ALFA,CDIPOL,CIMF,BIMF,DIR,
1      ISWIT,X,Y,Z,BXV,BYV,BZV)
C      IF (HJNEAR.NE.0.) THEN
C      CALL TAILC(ISWIT,NSTEP,X,Y,Z,RADIUS,DY,D,XNST,ZNST,BNST,
1      BXT,BYT,BZT)
C      IF (COLLAP.NE.1.) THEN
C      CALL TCOLLP(ISWIT,NSTEP,X,Y,Z,RADIUS,DYC,D,XNST,ZNST,
1      BNST,COLLAP,DELXC,BXTC,BYTC,BZTC)
C      END IF
C      END IF
C      IF ((BRN.NE.0.).OR.(BRP.NE.0.)) THEN
C      CALL RINGC(X,Y,Z,TILT,BRN,BRP,RN,RP,BXR,BYR,BZR)
C      END IF
C
C      BXTOT = BXV + BXT + BXTC + BXR
C      BYTOT = BYV + BYT + BYTC + BYR
C      BZTOT = BZV + BZT + BZTC + BZR
C
C      END IF
C
C      RETURN
C      END
C
C***** GETRAD
C
C      SUBROUTINE GETRAD(STAND,RADIUS)
C
C      PURPOSE-  GIVES MAGNETOTAIL RADIUS AS A FUNCTION OF THE STAND-OFF
C      DISTANCE.
C      INPUT-    STAND: THE STAND-OFF DISTANCE (RE)
C      OUTPUT-   RADIUS: OF THE MAGNETOTAIL (RE)
C      VARIABLES-A0,A1,A2: COEFFICIENTS
C      MAXRAD: UPPER LIMIT RADIUS ALLOWED
C      MINRAD: LOWER LIMIT RADIUS ALLOWED
C      MXSTND: UPPER LIMIT STAND-OFF DISTANCE
C      MNSTND: LOWER LIMIT STAND-OFF DISTANCE
C      NOTE: DERIVED BY MATCHING STAND-OFF DISTANCES 7,11,14 RE WITH
C      RADII 17,20,27 RE RESPECTIVELY
C
C      A0 = 29.1660
C      A1 = -3.3214
```

90/06/30
12:04:14

mapint.for

25

```
A2 = 0.2262
MAXRAD = 27.
MINRAD = 17.
MXSTND = 14.
MNSTND = 7.

C
C
IF (STAND.GT.MXSTND) THEN
  RADIUS = MAXRAD
  STAND = MXSTND
C
ELSE
  IF (STAND.LT.MNSTND) THEN
    RADIUS = MINRAD
    STAND = MNSTND
  C
  ELSE
    RADIUS = A0 + A1*STAND + A2*(STAND**2)
    END IF
  END IF
C
RETURN
END
C***** TAILC
SUBROUTINE TAILC(ISWIT,NSTEP,XGSM,YGSM,ZGSM,RADIUS,DY,D,XNST,ZNST,
1          BNST,BX,BY,BZ)
C
C
C PURPOSE- CALCULATE THE MAGNETIC FIELD AND CURRENTS DUE TO A
C           CROSSTAIL CURRENT SHEET. MULTIPLE CURRENT SEGMENTS ARE
C           JOINED EDGE TO EDGE. THE CURRENT STRENGTH AND POSITION
C           AS A FUNCTION OF XGSM ARE ARBITRARY AND PROVIDED.
C           A YGSM DEPENDENCE IS ALSO INCLUDED. INSPIRATION FOR
C           THIS MODEL CAN BE FOUND IN
C           TSYGANENKO & USMANOV, PLANETARY AND SPACE SCIENCE,
C           VOL. 30, PP. 985-998, 1982
C
C INPUT-    ISWIT: COUNTER USED IN SUBROUT. ZPOS
C           NSTEP: NUMBER OF CURRENT SEGMENTS COMPRISING THE TAIL
C           XGSM,YGSM,ZGSM: THE GEOCENTRIC SOLAR MAGNETOSPHERIC
C                       COORDINATE (EARTH RADII)
C           NOTE: ALL REFERENCES TO DISTANCE ARE MADE IN EARTH
C           RADII IN THE GSM COORDINATE SYSTEM.
C           RADIUS: (RE) OF MAGNETOTAIL
C           DY: CHAR. WIDTH FOR Y DEPENDENCE OF BX AND BZ
C           D: HALF-WIDTH OF CURRENT FILAMENTS
C           XNST,ZNST: LOCATION OF CURRENT SHEET IN GSM X AND Z
C           BNST: CURRENT STRENGTH AT XNST,ZNST
C OUTPUT-   BX,BY,BZ MAGNETIC FIELD COMPONENTS IN GSM (GAMMAS)
C           HJX,HJY,HJZ CURRENT DENSITY IN GSM (AMPS/METER**2)
C
C VARIABLES-XN,XF: NEAR AND FAR EDGES OF A CURRENT SEGMENT
C           ZN,ZF: NEAR AND FAR ZGSM POSITION OF CURRENT SEGMENT
C           BN,BF: VALUES OF B AT XN AND XF (REALLY CURRENT STRENGTH)
C           DELX: STEP SIZE = (XN-XF)
C           FY: VALUE OF Y-DEPENDENT FUNCTION (SEE SUBROUT. YYY)
C           FYP: FIRST DERIVATIVE OF Y-DEPENDENT FUNCTION ABOVE
C           DELB: =(BN-BF)
C           DD: = DELB/DELX
C           SLOPE: SLOPE OF A SEGMENT =(ZF-ZN)/(XF-XN)
C           RADIUS: RADIUS OF MAGNETOTAIL IN RE
C           FACTJ: FACTOR TO GET CURRENT IN UNITS AMPS/METER**2
C                   = (1/MAGNETIC PERM)*(TESLA/GAMMA)*(RE/METER)
C                   = (1/(4*PI*1E-07))*(1E-09)*(1/(6.370E+06))
C                   = 1.249E-10
C           ALL OTHER VARIABLES ARE PRESENT FOR CONVENIENCE AND HAVE
```

90/06/30
12:04:14

mapint.for

26

```
C      NO PARTICULAR PHYSICAL MEANINGS IN MOST CASES. MAP17210
C      MAP17220
C      SUBROUT.- YYY: RETURNS VALUE OF YGSM FUNCTIONAL DEPENDENCE AND THEM MAP17230
C      FIRST DERIVATIVE W.R.T. YGSM AT A GIVEN YGSM MAP17240
C      MAP17250
C      MAP17260
C      DIMENSION XNST(50),ZNST(50),BNST(50) MAP17270
C      FACTJ = 1.249E-10 MAP17280
C      MAP17290
C      BX = 0. MAP17300
C      BY = 0. MAP17310
C      BZ = 0. MAP17320
C      MAP17330
C      CALCULATE CONTRIBUTIONS FROM NSTEP CURRENT SEGMENTS TO B-FIELD MAP17340
C      DO 10 N = 1, NSTEP MAP17350
C      MAP17360
C      XN = XNST(N) MAP17370
C      XF = XNST(N+1) MAP17380
C      DELX = XN - XF MAP17390
C      ZN = ZNST(N) MAP17400
C      ZF = ZNST(N+1) MAP17410
C      SLOPE = (ZF - ZN) / (XF - XN) MAP17420
C      MAP17430
C      BN = BNST(N) MAP17440
C      BF = BNST(N+1) MAP17450
C      DELB = BN - BF MAP17460
C      MAP17470
C      DD = DELB/DELX MAP17480
C      QB = BN - DD*XN MAP17490
C      MAP17500
C      CONSTANTS INVOLVED IN INTEGRATION: A,B,C ARE COMMON TO BOTH MAP17510
C      THE BX AND BZ EXPRESSIONS, BY = 0 MAP17520
C      MAP17530
C      A = 1. + SLOPE**2 MAP17540
C      B = -2.*(XGSM + SLOPE*(ZGSM - ZN + SLOPE*XN)) MAP17550
C      C = XGSM**2 + (ZGSM - ZN + SLOPE*XN)**2 + D**2 MAP17560
C      BDX = -2. MAP17570
C      BDZ = -2.*SLOPE MAP17580
C      CDX = 2.*XGSM MAP17590
C      CDZ = 2.*(ZGSM - ZN + SLOPE*XN) MAP17600
C      MAP17610
C      TEST FOR PROPER INTEGRAL FORM, ONLY THE FORM FOR ABC > 0 HAVE MAP17620
C      BEEN INCLUDED. NO EXCEPTIONS FOUND TO DATE MAP17630
C      MAP17640
C      ABC = 4.*A*C - B**2 MAP17650
C      SABC = SQRT(ABC) MAP17660
C      IF (ABC.LE.0.) THEN MAP17670
C      GOTO 15 MAP17680
C      END IF MAP17690
C      MAP17700
C      SOME CONSTANTS FOR BX EXPRESSION MAP17710
C      MAP17720
C      CMX = -SLOPE*DD MAP17730
C      CNX = -SLOPE*QB + DD*(SLOPE*XN + ZGSM - ZN) MAP17740
C      CPX = (SLOPE*XN + ZGSM - ZN)*QB MAP17750
C      MAP17760
C      SOME CONSTANTS FOR BZ EXPRESSION MAP17770
C      MAP17780
C      CMZ = DD MAP17790
C      CNZ = BN - DD*(XN + XGSM) MAP17800
C      CPZ = -XGSM*QB MAP17810
C      MAP17820
C      SOME COMMON FACTORS FOR BOTH BX AND BZ MAP17830
C      MAP17840
```

90/06/30
12:04:14

mapint.for

27

```

      WLN = LOG((A*XN**2+B*XN+C)/(A*XF**2+B*XF+C))
      WATAN = ATAN((2*A*XN+B)/SABC) - ATAN((2*A*XF+B)/SABC)
C
C      THE BX FUNCTION
C
      CLNX = (A*CNX - B*CMX)/(2.*(A**2))
      CTANX = (2.*A*(A*CPX-C*CMX)-B*(A*CNX-B*CMX))/((A**2)*SABC)
      BX = BX + (CMX*DELX)/A + CLNX*WLN + CTANX*WATAN
C
C      THE BZ FUNCTION
C
      CLNZ = (A*CNZ - B*CMZ)/(2.*(A**2))
      CTANZ = (2.*A*(A*CPZ-C*CMZ)-B*(A*CNZ-B*CMZ))/((A**2)*SABC)
      BZ = BZ + (CMZ*DELX)/A + CLNZ*WLN + CTANZ*WATAN
C
10  CONTINUE
C
C      GET YGSM DEPENDENCE AND FIRST DERIVATIVE
      CALL YYY(YGSM,DY,RADIUS,FY,FYP)
C
C      NOW INCLUDE THE YGSM DEPENDENCE IN B-FIELD AND CURRENT
      BX = BX*FY
      BY = 0.
      BZ = BZ*FY
      GOTO 20
C
15  CONTINUE
      WRITE(6,99)
99  FORMAT(1H1,'WRONG EQUATION USED, OUTPUT SET TO ZERO')
      BX = 0.
      BY = 0.
      BZ = 0.
C
20  CONTINUE
      RETURN
      END
C
C***** HJSIZE
C
      FUNCTION HJSIZE(X,XNEAR,HJNEAR,HJFRAC)
C
C      PURPOSE- CALCULATE RELATIVE CURRENT STRENGTH ALONG MIDNIGHT
C                MERIDIAN AT A GIVEN XGSM POSITION. FUNCTION IS ARBITRARY
C      INPUT-    X: XGSM POSITION
C                XNEAR: TAIL CURRENT INNER EDGE DISTANCE (NEG. TAILWARD)
C                HJNEAR: HJSIZE AT XNEAR
C                HJFRAC: FRACTION OF HJNEAR FOUND AT -100 RE
C      OUTPUT-   HJSIZE: STRENGTH OF CROSSTAIL CURRENT
C      VARIABLES-BFLAT: X POSITION WHERE HJSIZE LEVELS OFF TO CONSTANT REM
C                NOTE: BFLAT IS SET AT -100. RE, AN AD HOC VALUE
C                EXPON: EXPONENTIAL DECREASE OF CURRENT DENSITY
C
      BFLAT = -100.
      EXPON = LOG(HJFRAC)/(BFLAT - XNEAR)
      IF (X.GT.XNEAR) THEN
        HJSIZE = 0.
        GOTO 10
      END IF
C
      IF (X.GT.(BFLAT)) THEN
        HJSIZE = HJNEAR*EXP(EXPON*(X - XNEAR))
      ELSE
        HJSIZE = HJNEAR*HJFRAC
      END IF

```

MAP17850
MAP17860
MAP17870
MAP17880
MAP17890
MAP17900
MAP17910
MAP17920
MAP17930
MAP17940
MAP17950
MAP17960
MAP17970
MAP17980
MAP17990
MAP18000
MAP18010
MAP18020
MAP18030
MAP18040
MAP18050
MAP18060
MAP18070
MAP18080
MAP18090
MAP18100
MAP18110
MAP18120
MAP18130
MAP18140
MAP18150
MAP18160
MAP18170
MAP18180
MAP18190
MAP18200
MAP18210
MAP18220
MAP18230
MAP18240
MAP18250
MAP18260
MAP18270
MAP18280
MAP18290
MAP18300
MAP18310
MAP18320
MAP18330
MAP18340
MAP18350
MAP18360
MAP18370
MAP18380
MAP18390
MAP18400
MAP18410
MAP18420
MAP18430
MAP18440
MAP18450
MAP18460
MAP18470
MAP18480

90/06/30
12:04:14

mapint.for

28

```
C
10  CONTINUE
    RETURN
    END
C
C***** ZPOS
C
SUBROUTINE ZPOS (ISWIT, TILT, XGSM, ZGSM)
C
C    PURPOSE-  CALCULATE ZGSM POSITION OF CURRENT SHEET IN
C               MIDNIGHT MERIDIAN
C    INPUT-    TILT: DIPOLE TILT ANGLE
C               ISWIT: COUNTER SO COEFFICIENTS ONLY CALCULATED ONCE
C               XGSM: COORDINATE OF INTEREST
C    OUTPUT-   ZGSM: CURRENT SHEET ELEVATION W.R.T. THE XGSM AXIS
C    VARIABLES- C1: XGSM WHERE CURVE LEVELS OFF, ZERO SLOPE
C               C2: XGSM WHERE CURVE LEAVES XSM AXIS
C               C3: ZGSM AT XGSM=C2
C               C4: ZGSM AT XGSM>=C1
C               C5: SLOPE OF XSM AXIS IN GSM FRAME AT C2
C               RTILT: DIPOLE TILT IN RADIANS
C               HINGE: CURRENT SHEET HINGE POSITION, SET = 9. RE
C                       AFTER GOSLING, JGR, 7093, 1986
C               A0,A1,A2,A3: COEFFICIENTS OF POLYNOMIAL FIT
C                        $Z = A0 + A1*X + A2*X**2 + A3*X**3$ 
C
C    IF (ISWIT.LE.2) THEN
C        CALCULATE COEFFICIENTS
C        HINGE = 9.
C        DEG = ACOS(-1.00)/180.
C        RTILT = DEG*TILT
C
C        C1 = 4. - HINGE*(1. + COS(RTILT))
C        C2 = -4.*COS(RTILT)
C        C3 = 4.*SIN(RTILT)
C        C4 = HINGE*SIN(RTILT)
C        C5 = -TAN(RTILT)
C
C        Q1 = 3.*(C1**2)*C2 - C2**3
C        Q2 = -2.*C1*C2 + C2**2
C        Q3 = C2 - C1
C        Q4 = C2**2 - C1**2
C
C        A3 = (2.*(C4-C3)+C5*Q3)/(3.*Q3*Q4+2.*(Q1-2.*(C1**3)))
C        A2 = (C5 - 3.*A3*Q4)/(2.*Q3)
C        A1 = -2.*A2*C1 - 3.*A3*(C1**2)
C        A0 = C3 - A2*Q2 + A3*Q1
C    END IF
C
C    INSIDE CURVE LIMIT USE XSM AXIS
C    IF (XGSM.GT.C2) THEN
C        ZGSM = XGSM*C5
C    END IF
C
C    BETWEEN XGSM C2 AND C1, FITTED CURVE IS USED
C    IF ((XGSM.LE.C2).AND.(XGSM.GE.C1)) THEN
C        ZGSM = A0 + A1*XGSM + A2*(XGSM**2) + A3*(XGSM**3)
C    END IF
C
C    OUTSIDE CURVE LIMIT USE LINE PARALLEL TO XGSM AXIS
C    IF (XGSM.LT.C1) THEN
C        ZGSM = C4
C    END IF
C
```

90/06/30
12:04:14

mapint.for

29

```

RETURN
END
C
C*****
SUBROUTINE YYY(Y,DY,RAD,FY,FYP)
C
C  PURPOSE-  CALCULATE YGSM DEPENDENCE AND FIRST DERIVATIVE
C             FY = 0 AT Y = RADIUS
C  INPUT-    Y: YGSM POSITION (EARTH RADII)
C             RAD: RADIUS OF MAGNETOTAIL
C             DY: CHARACTERISTIC WIDTH OF Y DEPENDENCE
C  OUTPUT-   FY: FUNCTIONAL VALUE OF YGSM DEPENDENCE AT Y
C             FYP: FIRST DERIVATIVE OF FY W.R.T Y
C
C  YEXP = -(Y/DY)**2
C  IF (YEXP.LT.(-40.)) THEN
C    FY = 0.
C    FYP = 0.
C    GOTO 10
C  END IF
C
C  PI = ACOS(-1.00)
C  CY = COS(PI*Y/(2.*RAD))
C  SY = SIN(PI*Y/(2.*RAD))
C
C  FY = CY*EXP(-(Y/DY)**2)
C  FYP = (-2.*Y*CY/(DY**2) - PI*SY/(2.*RAD))*EXP(-(Y/DY)**2)
10 CONTINUE
C
C  RETURN
C  END
C
C*****
SUBROUTINE RINGC(XGSM,YGSM,ZGSM,TILT,BRN,BRP,RN,RP,
1             BXGSM,BYGSM,BZGSM)
C
C  PURPOSE-  CALCULATE MAGNETIC FIELD AND CURRENT COMPONENTS OF RING
C             CURRENT TILTED WITH THE DIPOLE
C  INPUT-    XGSM,YGSM,ZGSM: GSM COORDINATES IN RE
C             TILT: TILT IN DEGREES OF CENTRAL AXIS OF RINGS, ALIGNED
C                   WITH DIPOLE AXIS. POSITIVE FOR N. HEMISPHERE SUMM.
C             BRN: NEGATIVE BZ CONTRIBUTION FOR ZERO TILT AT (0,0,0)
C                   FROM THE WESTWARD CURRENT
C             BRP: POSITIVE BZ CONTRIBUTION FROM EASTWARD CURRENT
C             RN:  CHARACTERISTIC RADIUS (IN RE) OF WESTWARD RING
C             RP:  CHARACTERISTIC RADIUS (IN RE) OF EASTWARD RING
C             BXGSM,BYGSM,BZGSM: MAGNETIC FIELD COMPONENTS OF RING
C                               IN GAMMAS
C  OUTPUT-   FACTJ: CONSTANT USED TO ADJUST UNITS OF CURRENT, SEE
C  VARIABLES-BR: SUBROUT. TAILC FOR DERIVATION
C             RHO,PHI,Z: CYLINDRICAL COORDINATE SYSTEM
C  SUBROUTINES- TRANS6: CHANGES GSM CARTESIAN TO SM CYLINDRICAL
C                   VTRAN6: CHANGES SM CYLINDRICAL VECTORS TO GSM SYSTEM
C
C             FACTOR TO MAKE CURRENT OUTPUT IN AMPS/M**2
C  FACTJ = 1.249E-10
C  PI = ACOS(-1.00)
C
C             CHANGE GSM TO SM CYLINDRICAL COORDINATES
C  CALL TRANS6(TILT,XGSM,YGSM,ZGSM,RHO,PHI,Z)
C
C             SCALE FACTOR (PUT Z AND RHO INTO UNITS OF CHAR. RADII)

```

MAP19130
MAP19140
MAP19150
MAP19160
MAP19170
MAP19180
MAP19190
MAP19200
MAP19210
MAP19220
MAP19230
MAP19240
MAP19250
MAP19260
MAP19270
MAP19280
MAP19290
MAP19300
MAP19310
MAP19320
MAP19330
MAP19340
MAP19350
MAP19360
MAP19370
MAP19380
MAP19390
MAP19400
MAP19410
MAP19420
MAP19430
MAP19440
MAP19450
MAP19460
MAP19470
MAP19480
MAP19490
MAP19500
MAP19510
MAP19520
MAP19530
MAP19540
MAP19550
MAP19560
MAP19570
MAP19580
MAP19590
MAP19600
MAP19610
MAP19620
MAP19630
MAP19640
MAP19650
MAP19660
MAP19670
MAP19680
MAP19690
MAP19700
MAP19710
MAP19720
MAP19730
MAP19740
MAP19750
MAP19760

90/06/30
17:04:14

mapint.for

30

```

      ZN = Z/RN
      RHON = RHO/RN
      ZP = Z/RP
      RHOP = RHO/RP
C
C      COMMON FACTORS
      AN = RHON**2 + ZN**2 + 4.
      BN = 2.*(ZN**2) - (RHON**2) + 8.
      AP = RHOP**2 + ZP**2 + 4.
      BP = 2.*(ZP**2) - (RHOP**2) + 8.
C
C      CYLINDRICAL BFIELD
      BRHO = 12.*BRN*RHON*ZN/SQRT(AN**5) + 12.*BRP*RHOP*ZP/SQRT(AP**5)
      BPHI = 0.
      BZ = 4.*BRN*BN/SQRT(AN**5) + 4.*BRP*BP/SQRT(AP**5)
C
C      CHANGE VECTOR QUANTITIES BACK TO GSM COORDINATES
      CALL VTRAN6 (BRHO, BPHI, BZ, PHI, TILT, BXGSM, BYGSM, BZGSM)
C
      RETURN
      END
C
C***** TRANS6
      SUBROUTINE TRANS6 (TILT, XGSM, YGSM, ZGSM, RHO, PHI, Z)
C
C      PURPOSE-  CHANGE GSM CARTESIAN TO SM CYLINDRICAL COORDINATES
C      INPUT-    TILT: DIPOLE TILT IN DEGREES
C               XGSM, YGSM, ZGSM: GSM COORDINATES
C      OUTPUT-   RHO, PHI, Z: CYLINDRICAL COORDINATES OF SM SYSTEM
C               UNITS (RE, RADIANS, RE)
C      VARIABLES-RTILT: TILT ANGLE IN RADIANS
C
      PI = ACOS(-1.00)
      RTILT = TILT*(PI/180.)
C
C      CHANGE GSM TO SM COORDINATES
      XSM = XGSM*COS(RTILT) - ZGSM*SIN(RTILT)
      YSM = YGSM
      ZSM = XGSM*SIN(RTILT) + ZGSM*COS(RTILT)
C
      RHO = SQRT(XSM**2 + YSM**2)
      IF (RHO.EQ.0.) THEN
         RHO = 0.000001
      END IF
      PHI = ACOS(XSM/RHO)
      IF (YSM.LT.0.) THEN
         PHI = PHI + 2.*(PI - PHI)
      END IF
      Z = ZSM
C
      RETURN
      END
C
C***** VTRAN6
      SUBROUTINE VTRAN6 (VRHO, VPHI, VZ, PHI, TILT, VXGSM, VYGSM, VZGSM)
C
C      PURPOSE-  CHANGE VECTOR QUANTITIES OF SM CYLINDRICAL TO
C               VECTORS OF GSM COORDINATES
C      INPUT-    VRHO, VPHI, VZ: RHO, PHI, Z VECTORS
C               PHI: ANGLE PHI IN RADIANS
C               TILT: DIPOLE TILT ANGLE IN DEGREES
C      OUTPUT-   VXGSM, VYGSM, VZGSM: GSM VECTOR COMPONENTS
```

mapint.for

```
C      VARIABLES-RTILT: DIPOLE TILT ANGLE IN RADIANS      MAP20410
C
C      PI = ACOS(-1.00)      MAP20420
C      RTILT = TILT*(PI/180.)      MAP20430
C
C      CHANGE CYLINDRICAL TO CARTESIAN (BOTH SM)      MAP20440
C      VXSM = VRHO*COS(PHI) - VPHI*SIN(PHI)      MAP20450
C      VYSM = VRHO*SIN(PHI) + VPHI*COS(PHI)      MAP20460
C      VZSM = VZ      MAP20470
C
C      CHANGE SM TO GSM      MAP20480
C      VXGSM = VXSM*COS(RTILT) + VZSM*SIN(RTILT)      MAP20490
C      VYGSM = VYSM      MAP20500
C      VZGSM = -VXSM*SIN(RTILT) + VZSM*COS(RTILT)      MAP20510
C
C      RETURN      MAP20520
C      END      MAP20530
C
C***** TCOLLP      MAP20540
C      SUBROUTINE TCOLLP (ISWIT,NSTEP,XGSM,YGSM,ZGSM,RADIUS,DYC,D,      MAP20550
C      1      XNST,ZNST,BNST,COLLAP,DELXC,BX,BY,BZ)      MAP20560
C
C      ***** SPECIAL COLLAPSE OF TAIL SIMUALTION, INTRODUCES LESSER      MAP20570
C      CURRENT IN OPPOSITE DIRECTION AT NEAR PART OF TAIL, ADDED TO      MAP20580
C      REGULAR CURRENT.      MAP20590
C      *****      MAP20600
C      PURPOSE- CALCULATE THE MAGNETIC FIELD AND CURRENTS DUE TO A      MAP20610
C      CROSSTAIL CURRENT SHEET. MULTIPLE CURRENT SEGMENTS ARE      MAP20620
C      JOINED EDGE TO EDGE. THE CURRENT STRENGTH AND POSITION      MAP20630
C      AS A FUNCTION OF XGSM ARE ARBITRARY AND PROVIDED.      MAP20640
C      A YGSM DEPENDENCE IS ALSO INCLUDED.      MAP20650
C
C      INPUT- ISWIT: COUNTER USED IN SUBROUT. ZPOS      MAP20660
C      NSTEP: NUMBER OF CURRENT SEGMENTS IN TAIL      MAP20670
C      XGSM,YGSM,ZGSM: THE GEOCENTRIC SOLAR MAGNETOSPHERIC      MAP20680
C      COORDINATE (EARTH RADII)      MAP20690
C      NOTE: ALL REFERENCES TO DISTANCE ARE MADE IN EARTHMAP20700
C      RADII IN THE GSM COORDINATE SYSTEM.      MAP20710
C      DYC: CHAR. WIDTH FOR Y DEPENDENCE OF BX AND BZ      MAP20720
C      D: HALF-WIDTH OF CURRENT FILAMENTS      MAP20730
C      XNST,ZNST: XGSM AND ZGSM POSITION OF CURRENT SHEET      MAP20740
C      BNST: CURRENT STRENGTH AT ANST,ZNST      MAP20750
C      COLLAP: FRACTION OF AIL FIELD CURRENT LIET AT MIDNIGHT      MAP20760
C      = 1. MEANS NO CHANGE, ORIGINAL TAIL ONLY      MAP20770
C      = 0. MEANS COMPLETE COLLAPSE AT INNER EDGE      MAP20780
C      DELXC: LENGTH OF INNER TAIL CURRENT SHEET (IN X) THAT ISMAP20790
C      DISRUPTED      MAP20800
C
C      OUTPUT- BX,BY,BZ MAGNETIC FIELD COMPONENTS IN GSM (GAMMAS)      MAP20810
C
C      VARIABLES-XN,XF: NEAR AND FAR EDGES OF A CURRENT SEGMENT      MAP20820
C      BN,BF: VALUES OF B AT XN AND XF (REALLY CURRENT STRENGTH)      MAP20830
C      NSTEP: NUMBER OF CURRENT SEGMENTS IN TAIL WITHIN 100 RE      MAP20840
C      DELX: STEP SIZE = (XN-XF)      MAP20850
C      FY: VALUE OF Y-DEPENDENT FUNCTION (SEE SUBROUT. YYY )      MAP20860
C      FYP: FIRST DERIVATIVE OF Y-DEPENDENT FUNCTION ABOVE      MAP20870
C      DELB: =(BN-BF)      MAP20880
C      DD: = DELB/DELX      MAP20890
C      SLOPE: SLOPE OF A SEGMENT =(ZF-ZN)/(XF-XN)      MAP20900
C      = 1.249E-10      MAP20910
C      ALL OTHER VARIABLES ARE PRESENT FOR CONVENIENCE AND HAVE      MAP20920
C      NO PARTICULAR PHYSICAL MEANINGS IN MOST CASES.      MAP20930
C      MAP20940
C      MAP20950
C      MAP20960
C      MAP20970
C      MAP20980
C      MAP20990
C      MAP21000
C      MAP21010
C      MAP21020
C      MAP21030
C      MAP21040
```

90/06/30
12:04:14

mapint.for

32

SUBROUT.- YYY: RETURNS YGSM FUNCTIONAL DEPENDENCE AND THE
FIRST DERIVATIVE W.R.T. YGSM AT A GIVEN YGSM

DIMENSION XNST(50),ZNST(50),BNST(50)

BX = 0.
BY = 0.
BZ = 0.

CALCULATE CONTRIBUTIONS FROM NSTEP CURRENT SEGMENTS TO B-FIELD

DO 10 N = 1, NSTEP

XN = XNST(N)
XF = XNST(N+1)
DELX = XN - XF

ZN = ZNST(N)
ZF = ZNST(N+1)
SLOPE = (ZF - ZN) / (XF - XN)

BN = BNST(N)
BF = BNST(N+1)

ADJUST CURRENT SIZE AND DIRECTION FROM ORIGINAL
(0.1) FACTOR BELOW GIVES 5 RE COLLAPSE "EDGE"
CFN = .5*(TANH(0.1*(-XN+XNST(1)-DELXC))-TANH(-0.1*DELXC) - 2.)
CFF = .5*(TANH(0.1*(-XF+XNST(1)-DELXC))-TANH(-0.1*DELXC) - 2.)

NO NEED TO CALCULATE COLLAPSE CURRENT FOR FAR TAIL
CUT OUT OF LOOP WHEN CONTRIBUTION IS LESS THAN 2 PERCENT
THE MAGNITUDE OF THE REGULAR CURRENT OF TAIL
IF (CFF.GT.(-0.02)) THEN

GOTO 11
END IF
BN = (1.- COLLAP)*BN*CFN
BF = (1.- COLLAP)*BF*CFF

DELB = BN - BF
DD = DELB/DELX
QB = BN - DD*XN

CONSTANTS INVOLVED IN INTEGRATION: A,B,C ARE COMMON TO BOTH
THE BX AND BZ EXPRESSIONS, BY = 0

A = 1. + SLOPE**2
B = -2.*(XGSM + SLOPE*(ZGSM - ZN + SLOPE*XN))
C = XGSM**2 + (ZGSM - ZN + SLOPE*XN)**2 + D**2
BDX = -2.
BDZ = -2.*SLOPE
CDX = 2.*XGSM
CDZ = 2.*(ZGSM - ZN + SLOPE*XN)

TEST FOR PROPER INTEGRAL FORM, ONLY THE FORM FOR ABC > 0 HAVE
BEEN INCLUDED. NO EXCEPTIONS FOUND TO DATE
ABC = 4.*A*C - B**2
SABC = SQRT(ABC)
IF (ABC.LE.0.) THEN
GOTO 15
END IF

SOME CONSTANTS FOR BX EXPRESSION
CMX = -SLOPE*DD
CNX = -SLOPE*QB + DD*(SLOPE*XN + ZGSM - ZN)
CPX = (SLOPE*XN + ZGSM - ZN)*QB

MAP21050
MAP21060
MAP21070
MAP21080
MAP21090
MAP21100
MAP21110
MAP21120
MAP21130
MAP21140
MAP21150
MAP21160
MAP21170
MAP21180
MAP21190
MAP21200
MAP21210
MAP21220
MAP21230
MAP21240
MAP21250
MAP21260
MAP21270
MAP21280
MAP21290
MAP21300
MAP21310
MAP21320
MAP21330
MAP21340
MAP21350
MAP21360
MAP21370
MAP21380
MAP21390
MAP21400
MAP21410
MAP21420
MAP21430
MAP21440
MAP21450
MAP21460
MAP21470
MAP21480
MAP21490
MAP21500
MAP21510
MAP21520
MAP21530
MAP21540
MAP21550
MAP21560
MAP21570
MAP21580
MAP21590
MAP21600
MAP21610
MAP21620
MAP21630
MAP21640
MAP21650
MAP21660
MAP21670
MAP21680

90/06/30
12:04:14

33

mapint.for

```
C      SOME  CONSTANTS FOR BZ EXPRESSION                                MAP21690
CMZ = DD                                                                MAP21700
CNZ = BN - DD*(XN + XGSM)                                              MAP21710
CPZ = -XGSM*QB                                                         MAP21720
C                                                                           MAP21730
C      SOME  COMMON FACTORS FOR BOTH BX AND BZ                         MAP21740
C                                                                           MAP21750
WLN = LOG((A*XN**2+B*XN+C)/(A*XF**2+B*XF+C))                          MAP21760
WATAN = ATAN((2*A*XN+B)/SABC) - ATAN((2*A*XF+B)/SABC)                 MAP21770
C                                                                           MAP21780
C      THE  BX FUNCTION                                                MAP21790
CLNX = (A*CNX - B*CMX)/(2.*(A**2))                                    MAP21800
CTANX = (2.*A*(A*CPX-C*CMX)-B*(A*CNX-B*CMX))/((A**2)*SABC)           MAP21810
BX = BX + (CMX*DELX)/A + CLNX*WLN + CTANX*WATAN                       MAP21820
C                                                                           MAP21830
C      THE  BZ FUNCTION                                                MAP21840
CLNZ = (A*CNZ - B*CMZ)/(2.*(A**2))                                    MAP21850
CTANZ = (2.*A*(A*CPZ-C*CMZ)-B*(A*CNZ-B*CMZ))/((A**2)*SABC)           MAP21860
BZ = BZ + (CMZ*DELX)/A + CLNZ*WLN + CTANZ*WATAN                       MAP21870
C                                                                           MAP21880
10  CONTINUE                                                            MAP21890
11  CONTINUE                                                            MAP21900
C                                                                           MAP21910
C      USING SAME Y FUNCTION AS TAILC SUBROUT., BUTWITH DIFFERENT      MAP21920
C      DY VALUE (DYC HERE)                                              MAP21930
C      GET YGSM DEPENDENCE AND FIRST DERIVATIVE                       MAP21940
CALL YYY(YGSM,DYC,RADIUS,FY,FYP)                                       MAP21950
C                                                                           MAP21960
C      NOW INCLUDE THE YGSM DEPENDENCE IN B-FIELD AND CURRENT          MAP21970
BX = BX*FY                                                              MAP21980
BY = 0.                                                                  MAP21990
BZ = BZ*FY                                                              MAP22000
GOTO 20                                                                  MAP22010
C                                                                           MAP22020
15  CONTINUE                                                            MAP22030
WRITE(6,99)                                                             MAP22040
99  FORMAT(1H1,'WRONG EQUATION USED, OUTPUT SET TO ZERO')              MAP22050
BX = 0.                                                                  MAP22060
BY = 0.                                                                  MAP22070
BZ = 0.                                                                  MAP22080
C                                                                           MAP22090
20  CONTINUE                                                            MAP22100
RETURN                                                                    MAP22110
END                                                                      MAP22120
C***** SEGMENT                                                         MAP22130
SUBROUTINE SEGMENT(ISWIT,TILT,XNEAR,HJNEAR,HJFRAC,NSTEP,XNST,          MAP22140
1  ZNST,BNST)                                                           MAP22150
C                                                                           MAP22160
C      PURPOSE-  FIND THE GSM X AND Z LOCATIONS OF THE SEGMENT BOUNDARIES MAP22170
C                USED TO BUILD THE TAIL CURRENT. ALSO ASSIGN A CURRENT  MAP22180
C                STRENGTH FOR EACH POINT.  THE INNER EDGE IS ABS(XNEAR)  MAP22190
C                AWAY FROM EARTH.                                         MAP22200
C                                                                           MAP22210
C      INPUT-    ISWIT: COUNTER USED IN SUBROUT. ZPOS                   MAP22220
C                TILT: DIPOLE TILT IN DEGREES                           MAP22230
C                XNEAR: NEGATIVE DISTANCE TAILWARD, LOCATION OF INNER EDGE MAP22240
C                HJNEAR: CURRENT STRENGTH AT INNER EDGE                 MAP22250
C                HJFRAC: FRACTION OF HJNEAR REMAINING IN FAR TAIL        MAP22260
C      OUTPUT-   NSTEP: NUMBER OF SEGMENTS USED TO BUILD THE TAIL CURRENT MAP22270
C                XNST,ZNST: GSM X AND Z LOCATIONS SEGMENT BOUNDARIES    MAP22280
C                BNST: CURRENT STRENGTH AT (XNST,ZNST)                  MAP22290
C      VARIABLES-FRAC: FRACTION OF DESIRED SEGMENT LENGTH               MAP22300
C                PI: JUST PI                                             MAP22310
C                RTILT: TILT ANGLE IN RADIANS                            MAP22320
```

90/06/30
12:04:14

34

mapint.for

```
C      XNGSM,ZNGSM: TEST XGSM AND ZGSM TO FIND INNER EDGE      MAP22330
C      SN: DISTANCE ALONG SHEET PLUS INNER EDGE DISTANCE      MAP22340
C      DELS: DESIRED SEGMENT LENGTH      MAP22350
C      DIST: TEST DISTANCE COMPARED TO DELS OR ABS(XNEAR)      MAP22360
C      MAP22370
C      SUBROUT.- ZPOS: FINDS ZGSM LOCATION OF CURRENT SHEET AT ANY XGSM      MAP22380
C      HJSIZE: FINDS CURRENT STRENGTH AT POINT ALONG SHEET      MAP22390
C      MAP22400
C      DIMENSION XNST(50),ZNST(50),BNST(50)      MAP22410
C      MAP22420
C      NSTEP = 16      MAP22430
C      TAIL IS 350 RE LONG, MADE OF 16 SEGMENTS OF VARYING LENGTHS      MAP22440
C      FRAC = 0.025      MAP22450
C      SET INNER EDGE POINT, ABS(XNEAR) IS DISTANCE FROM EARTH      MAP22460
C      IF (TILT.NE.0.) THEN      MAP22470
C      STEP ALONG XGSM AXIS UNTIL PROPER DISTANCE IS FOUND      MAP22480
C      PI = ACOS(-1.00)      MAP22490
C      RTILT = TILT*PI/180      MAP22500
C      XNGSM = 0.0      MAP22510
C      2 CONTINUE      MAP22520
C      XNGSM = XNGSM - 0.05      MAP22530
C      CALL ZPOS(ISWIT,TILT,XNGSM,ZNGSM)      MAP22540
C      DIST = SQRT(XNGSM**2 + ZNGSM**2)      MAP22550
C      SEE IF INNER EDGE IS AT PROPER DISTANCE FROM EARTH      MAP22560
C      IF (DIST.GE.ABS(XNEAR)) THEN      MAP22570
C      XNST(1) = XNGSM      MAP22580
C      ZNST(1) = ZNGSM      MAP22590
C      ELSE      MAP22600
C      GOTO 2      MAP22610
C      END IF      MAP22620
C      ELSE      MAP22630
C      TILT = 0      MAP22640
C      XNST(1) = XNEAR      MAP22650
C      ZNST(1) = 0.      MAP22660
C      END IF      MAP22670
C      SN = XNST(1)      MAP22680
C      BNST(1) = HJSIZE(SN,XNST(1),HJNEAR,HJFRAC)      MAP22690
C      GET OTHER SEGMENT BOUNDARY POINTS      MAP22700
C      DO 10 N = 1,NSTEP      MAP22710
C      I = N + 1      MAP22720
C      GET STEP SIZE      MAP22730
C      IF (N.LE.8) THEN      MAP22740
C      DELS = 2.0      MAP22750
C      ELSE      MAP22760
C      IF (N.LE.11) THEN      MAP22770
C      DELS = 5.      MAP22780
C      ELSE      MAP22790
C      IF (N.LE.14) THEN      MAP22800
C      DELS = 10.      MAP22810
C      ELSE      MAP22820
C      IF (N.EQ.15) THEN      MAP22830
C      DELS = 39.      MAP22840
C      END IF      MAP22850
C      END IF      MAP22860
C      END IF      MAP22870
C      END IF      MAP22880
C      IF (N.EQ.NSTEP) THEN      MAP22890
C      DELS = 250.      MAP22900
C      END IF      MAP22910
C      MAP22920
C      SN = SN - DELS      MAP22930
C      BNST(I) = HJSIZE(SN,XNST(1),HJNEAR,HJFRAC)      MAP22940
C      IF (TILT.NE.0.) THEN      MAP22950
C      IF (I.GT.2) THEN      MAP22960
```

90/06/30
12:04:14

35

mapint.for

```
C      IF (ZNST(I-1).EQ.ZNST(I-2)) THEN
      NOW IN FLAT PART OF CURRENT SHEET
      XNST(I) = XNST(I-1) - DELS
      ZNST(I) = ZNST(I-1)
      GOTO 9
      END IF
      END IF
      XNST(I) = XNST(I-1)
5     CONTINUE
      XNST(I) = XNST(I) - FRAC*DELS
      CALL ZPOS(ISWIT,TILT,XNST(I),ZNST(I))
      DIST = SQRT((XNST(I)-XNST(I-1))**2+(ZNST(I)-ZNST(I-1))**2)
      IF (DIST.GE.DEELS) THEN
C      THEN NEXT SEGMENT BOUNDARY IS FOUND
      GOTO 9
      ELSE
C      ADD ANOTHER PIECE
      GOTO 5
      END IF
      ELSE
C      TILT IS ZERO
      XNST(I) = SN
      ZNST(I) = 0.
      END IF
9     CONTINUE
10    CONTINUE
      RETURN
      END
C *****: BVOIGT
      SUBROUTINE BVOIGT(STAND,RADIUS,TILT,ALFA,CDIPOL,CIMF,BIMF,DIR,
1      ISWIT,X,Y,Z,BX,BY,BZ)
C
C  PURPOSE-  COMPUTES THREE COMPONENTS OF THE VECTOR MAGNETIC FIELD
C            AT ANY POINT. POINTS OUTSIDE OF THE MAGNETOPAUSE WILL
C            ONLY RETURN NONZERO FIELDS IF INTERCONNECTION IS ALLOWED
C            WITH THE IMF.
C
C  INPUT-    STAND, THE SUBSOLAR STAND-OFF DISTANCE (IN EARTH RADII)
C            RADIUS, RADIUS OF THE MAGNETOTAIL (IN EARTH RADII)
C            TILT, DIPOLE TILT ANGLE (IN DEGREES)
C            TILT>0 INDICATES NORTHERN HEMISPHERE SUMMER
C            ALFA, PARAMETER INDICATING PLASMA CONTENT OF TAIL,
C            = 1.0 MEANS VACUUM, = 0.0 MEANS HARRIS SHEET
C            CDIPOL, FRACTION OF DIPOLE FIELD ALLOWED TO PENETRATE
C            OUT THROUGH THE MAGNETOPAUSE
C            CIMF, FRACTION OF THE INTERPLANETARY FIELD ALLOWED TO
C            PENETRATE IN THROUGH THE MAGNETOPAUSE
C            BIMF, MAGNITUDE OF THE INTERPLANETARY MAGNETIC FIELD
C            (IN GAMMA)
C            DIR, DIRECTION OF IMF, 0 DEGREES IS NORTH, 90 DEGREES
C            IS ANTISUNWARD
C            ISWIT, INDICATES THE PROCEDURE TO FOLLOW ON THIS CALL
C            BFIELD IS CALLED THREE TIMES WITH ISWIT = 1,2
C            THEN EITHER 3 OR 4
C            *****ALTERED SO >2 GIVES GSM RESULT 10/11/87
C            = 1 GENERAL COEFFICIENTS ARE FOUND TO REPRESENT
C            THE MAGNETIC FIELD WITH LEGENDRE AND BESSEL
C            SERIES
C            = 2 CONSTANTS SPECIFY THE CONFIGURATION DESIRED
C            ARE CALCULATED AND STORED, I.E. TILT
C            = 3 CALCULATES FIELD COMPONENTS, INPUT AND OUTPUT
C            ARE IN THE SM (SOLAR MAGNETOSPHERIC) SYSTEM
C            = 4 SAME AS 3 BUT GSM COORDINATES ARE USED
C            X,Y,Z CARTESIAN COORDINATE IN EITHER SM OR GSM SYSTEM
```


mapint.for

```
C
C      OUTPUT-   BX,BY,BZ   MAGNETIC FIELD VECTOR IN CARTESIAN SYSTEM
C                  ACCORDING TO CHOICE ISWIT=3 OR 4. (IN GAMMA)
C
C
C      VARIABLES- NDIM   NUMBER OF GRID POINTS TO BE USED FOR INTEGRATION
C                  TO OBTAIN SERIES COEFFICIENTS
C                  NELE   NUMBER OF ITERATIONS TO BE MADE BETWEEN SPHERICAL
C                  AND CYLINDRICAL COEFFICIENT CALCULATIONS.
C                  BEX    DISTANCE BETWEEN SPHERE CENTER AND EARTH
C                  XD,YD,ZD  CARTESIAN COORDINATES IN DIPOLE SYSTEM, XD IS
C                  PERPEND. TO DIPOLE MOMENT POINTING ANTISUNWARD
C                  HXD,HYD,HZD  MAGNETIC FIELD COMPONENTS IN DIPOLE SYSTEM
C                  XM,YM,ZM   EARTH CENTERED CARTESIAN SYSTEM, XM POINTS ANTI-
C                  SUNWARD, PARALLEL TO THE TAIL AXIS (CALLED THE M
C                  SYSTEM)
C                  HXM,HYM,HZM  MAGNETIC FIELD COMPONENTS ON M SYSTEM
C                  SNPS,CSPS  SIN AND COS OF THE DIPOLE TILT ANGLE
C
C      REFERENCES-
C
C          REF1:          VOIGT,G.-H., A MATHEMATICAL MAGNETOSPHERIC
C          FIELD MODEL WITH INDEPENDENT PHYSICAL PARAMETERS, PLANETARY
C          AND SPACE SCIENCE, VOL. 29, 1-20,1981
C
C          REF2:          ABRAMOWITZ AND STEGUN,EDS., HANDBOOK OF
C          MATHEMATICAL FUNCTIONS, U.S. NATIONAL BUREAU OF STANDARDS,
C          DOVER, NEW YORK, (1965)
C
C          REF3:          JAHNKE-EMDE-LOSCH, TABLES OF HIGHER
C          FUNCTIONS, 6TH EDITION(REVISED), NEW YORK, (1960)
C
C          REF4:          VOIGT PH.D. THESIS
C
C      COORDINATE SYSTEMS-
C
C          GSM          GEOCENTRIC SOLAR MAGNETOSPHERIC - EARTH CENTERED
C
C                  XGSM : FROM EARTH TO SUN
C                  YGSM : PERPENDICULAR TO EARTH'S DIPOLE, POINTING
C                  TOWARDS DUSK
C                  ZGSM : SAME SENSE AS NORTHERN MAGNETIC POLE
C                  DIPOLE CONTAINED WITHIN X-Z PLANE
C
C          SM           SOLAR MAGNETIC - EARTH CENTERED
C
C                  XSM : TOWARD SUN BUT NOT DIRECTLY UNLESS TILT ZERO
C                  YSM = YGSM
C                  ZSM : PARALLEL TO NORTH MAGNETIC POLE
C
C          M            THE M SYSTEM (XM,YM,ZM) - EARTH CENTERED
C
C                  XM = -XGSM
C                  YM = -YGSM
C                  ZM =  ZGSM
C
C          V            THE V SYSTEM (XV,YV,ZV) - SPHERE CENTERED
C
C                  XV = ZGSM
C                  YV = YGSM
C                  ZV = -XGSM - B  WHERE B = (TAIL RADIUS)-(STAND-OFF)
C
C          D            THE D SYSTEM (XD,YD,ZD) - EARTH CENTERED
```

```
MAP23610
MAP23620
MAP23630
MAP23640
MAP23650
MAP23660
MAP23670
MAP23680
MAP23690
MAP23700
MAP23710
MAP23720
MAP23730
MAP23740
MAP23750
MAP23760
MAP23770
MAP23780
MAP23790
MAP23800
MAP23810
MAP23820
MAP23830
MAP23840
MAP23850
MAP23860
MAP23870
MAP23880
MAP23890
MAP23900
MAP23910
MAP23920
MAP23930
MAP23940
MAP23950
MAP23960
MAP23970
MAP23980
MAP23990
MAP24000
MAP24010
MAP24020
MAP24030
MAP24040
MAP24050
MAP24060
MAP24070
MAP24080
MAP24090
MAP24100
MAP24110
MAP24120
MAP24130
MAP24140
MAP24150
MAP24160
MAP24170
MAP24180
MAP24190
MAP24200
MAP24210
MAP24220
MAP24230
MAP24240
```

90/06/30
12:04:14

mapint.for

37

```
C          XD = -XSM                      MAP24250
C          YD = -YSM = -YGSM             MAP24260
C          ZD = ZSM                      MAP24270
C                                         MAP24280
C          SPHERICAL SYSTEM (R,THETA,PHI) - SPHERE CENTERED MAP24290
C                                         MAP24300
C          R : DISTANCE FROM ORIGIN (XV,YV,ZV)=(0,0,0) MAP24310
C          THETA : ANGLE BETWEEN ZV AND VECTOR R MAP24320
C          PHI : ANGLE BETWEEN XV AND COMPONENT OF VECTOR R MAP24330
C                  IN THE XV-YV PLANE MAP24340
C                                         MAP24350
C          CYLINDRICAL SYSTEM (RHO,PHI,Z) - SPHERE CENTERED MAP24360
C                                         MAP24370
C          RHO : RADIAL DISTANCE FROM ZV AXIS MAP24380
C          PHI : ANGLE BETWEEN XV-ZV PLANE AND VECTOR RHO MAP24390
C                  (SAME AS PHI OF SPHERICAL) MAP24400
C          Z = ZV MAP24410
C                                         MAP24420
C          COMMON /BLOCK1/ BEX,R,Q,PI,PSI MAP24430
C          COMMON /BLOCK3/ H0, ALFB, N MAP24440
C          COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS MAP24450
C          COMMON /BLOCK6/ CT, ST MAP24460
C          COMMON /FIMF/ HIMF, CD, CI, XI MAP24470
C                                         MAP24480
C          ALFB = ALFA MAP24490
C                                         MAP24500
C          R = RADIUS MAP24510
C          CD = CDIPOL MAP24520
C          CI = CIMF MAP24530
C          HIMF = BIMF MAP24540
C          XI = DIR MAP24550
C                                         MAP24560
C                                         MAP24570
C          IF (ISWIT.EQ.1) THEN MAP24580
C              GET COEFFICIENTS FOR SPHERICAL AND BESSEL HARMONICS MAP24590
C              BEX = RADIUS - STAND MAP24600
C              NDIM = 101 MAP24610
C              NELE = 4 MAP24620
C              CALL COEFS (NDIM,NELE) MAP24630
C          END IF MAP24640
C                                         MAP24650
C          IF (ISWIT.EQ.2) THEN MAP24660
C              SET CONSTANTS, SPECIFY TILT WHICH WAS PRESET IN COEFS MAP24670
C              NOTE THAT SIGN OF TILT ANGLE IS BEING SWITCH MAP24680
C              TI = - TILT MAP24690
C              CALL CONST(TI) MAP24700
C          END IF MAP24710
C                                         MAP24720
C          SM COORDINATE OPTION CUTOFF FOR NOW EVERYTHING IN GSM MAP24730
C          IF (ISWIT.EQ.3) THEN MAP24740
C              GET BFIELD AT POINT USING SM COORDINATE SYSTEM MAP24750
C              TRANSFORM SM TO DIPOLE SYSTEM, XD IS PERPENDICULAR MAP24760
C              TO DIPOLE MOMENT POINTING OPPOSITE XSM MAP24770
C          CCC XD = - X MAP24780
C          CCC YD = - Y MAP24790
C          CCC ZD = + Z MAP24800
C                                         MAP24810
C              TRANSFORM DIPOLE TO M SYSTEM, XM ANTISOLAR MAP24820
C          CCC CALL TRANS4 (XD,YD,ZD,XM,YM,ZM) MAP24830
C                                         MAP24840
C              GET FIELD COMPONENTS AT XM,YM,ZM MAP24850
C          CCC CALL BTOTAL (XM,YM,ZM,HXM,HYM,HZM) MAP24860
C                                         MAP24870
C              TRANSFORM FIELD COMPONENTS BACK TO DIPOLE SYSTEM MAP24880
```

90/06/30
12:04:14

mapint.for

38

```
CCC      HXD = HXM*CSPPS - HZM*SNPS      MAP24890
CCC      HYD = HYM      MAP24900
CCC      HZD = HXM*SNPS + HZM*CSPPS      MAP24910
C      MAP24920
C      OUTPUT IN SM COORDINATES (GAMMAS)      MAP24930
CCC      BX = - HXD      MAP24940
CCC      BY = - HYD      MAP24950
CCC      BZ = + HZD      MAP24960
C      MAP24970
CCC      END IF      MAP24980
C      MAP24990
C      ALL CALLS WITH ISWIT > 2 GIVE POINT RESULT IN GSM COORDINATES      MAP25000
C      IF (ISWIT.GT.2) THEN      MAP25010
C      GET BFIELD AT POINT USING GSM COORDINATE SYSTEM      MAP25020
C      XM = - X      MAP25030
C      YM = - Y      MAP25040
C      ZM = + Z      MAP25050
C      MAP25060
C      GET BFIELD COMPONENTS AT XM,YM,ZM      MAP25070
C      CALL BTOTAL(XM,YM,ZM,HXM,HYM,HZM)      MAP25080
C      MAP25090
C      OUTPUT IN GSM COORDINATES (GAMMAS)      MAP25100
C      BX = - HXM      MAP25110
C      BY = - HYM      MAP25120
C      BZ = + HZM      MAP25130
C      MAP25140
C      END IF      MAP25150
C      RETURN      MAP25160
C      END      MAP25170
C *****:      COEFFS      MAP25180
C      SUBROUTINE COEFFS (NDIM,NELE)      MAP25190
C      MAP25200
C      PURPOSE- CALCULATE COEFFICIENTS      MAP25210
C      INPUT- NDIM, NUMBER OF GRID POINTS FOR NUMERICAL INTEGRATION      MAP25220
C      NELE, NUMBER OF INTERATIONS TO OBTAIN COEFFICIENTS      MAP25230
C      MAP25240
C      FILL COMMON BLOCK SITES WITH BESSEL ROOTS AND VALUES      MAP25250
C      CALL ZERO      MAP25260
C      MAP25270
C      DO 10 KSA = 1, NELE      MAP25280
C      GET COEFFICIENTS OF LEGENDRE POLYNOMIALS      MAP25290
C      CALL SPHRCO (NDIM,KSA)      MAP25300
C      GET COEFFICIENTS OF BESSEL SERIES      MAP25310
C      CALL TAILCO (NDIM)      MAP25320
10  CONTINUE      MAP25330
C      MAP25340
C      RETURN      MAP25350
C      END      MAP25360
C *****:      CONST      MAP25370
C      SUBROUTINE CONST (TI)      MAP25380
C      MAP25390
C      PURPOSE- PROVIDE CONSTANTS (TO COMMON BLOCKS)      MAP25400
C      INPUT- TI, DIPOLE TILT ANGLE IN DEGREES      MAP25410
C      TI>0 INDICATES NORTHERN HEMISPHERE WINTER      MAP25420
C      R, RADIUS OF TAIL (FROM BLOCK 1) IN EARTH RADII      MAP25430
C      OUTPUT- Q, RATIO OF DIPOLE MOMENT/(RADIUS**3) IN GAMMA      MAP25440
C      PSI, DIPOLE TILT ANGLE IN RADIANS      MAP25450
C      PI, RADIAN EQUIVALENT OF 180 DEGREES      MAP25460
C      SNPS AND CSPPS, SIN AND COS OF PSI      MAP25470
C      VARIABLES- DM, DIPOLE MOMENT IN GAMMA* (METERS**3)      MAP25480
C      RE, EARTH RADIUS, METERS      MAP25490
C      MAP25500
C      MAP25510
C      COMMON /BLOCK1/ BEX,R,Q,PI,PSI      MAP25520
```

00/06/30
12:04:14

mapint.for

39

```
COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
COMMON /BLOCK6/ CT, ST
C
C   PI = ACOS(-1.00)
C
C   DM = 7.951E+24
C   RE = 6.37104E+06
C   ZA = R*RE
C   Q  = DM/(ZA**3)
C
C   PSI = (PI*TI)/180.0
C   CSPS = COS(PSI)
C   SNPS = SIN(PSI)
C
C   RETURN
C   END
C *****: BTOTAL
C   SUBROUTINE BTOTAL(XM,YM,ZM,HXM,HYM,HZM)
C
C   PURPOSE- GET FIELD COMPONENTS USING PREDETERMINED COEFFICIENTS
C   INPUT-   XM,YM,ZM M COORDINATES
C   OUTPUT-  HXM,HYM,HZM FIELD COMPONENTS IN M SYSTEM IN GAMMAS
C
C   COMMON /BLOCK1/ BEX,R,Q,PI,PSI
C   COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
C   COMMON /BLOCK6/ CT, ST
C
C   TRANSFORM M TO SPHERICAL COORDINATES RR,T,PHI
C   TWO STEP TRANSFORM BECAUSE ZV IS NEEDED BELOW IN 'IF'
C   CALL TRANS5(XM,YM,ZM,XV,YV,ZV,D1,D2,D3)
C   CALL TRANS3(XV,YV,ZV,RR,T,PHI)
C
C   IF(ZV.LT.0.0) THEN
C       GET FIELD FROM DAYSIDE OF SPHERE ZV<0
C       CALL DAYTOT(RR,T,PHI,HR,HT,HP)
C       TRANSFORM TO CARTESIAN ZV DOWN TAIL
C       CALL BTRAN1(HR,HT,HP,HXV,HYV,HZV)
C   ELSE
C       GET FIELD FROM NIGHTSIDE OF SPHERE ZV>0
C       RRZ = SQRT(XV*XV + YV*YV)
C       CALL TAILTO(RRZ,PHI,ZV,HR,HP,HZ)
C       TRANSFORM TO CARTESIAN ZV DOWN TAIL
C       CALL BTRAN2(HR,HZ,HP,HXV,HYV,HZV)
C   END IF
C
C   TRANSFORM BACK TO M SYSTEM FOR OUTPUT
C   HXM = + HZV
C   HYM = - HYV
C   HZM = + HXV
C
C   RETURN
C   END
C *****: TRANS4
C   SUBROUTINE TRANS4(XD,YD,ZD,XM,YM,ZM)
C
C   PURPOSE- TRANSFORM DIPOLE TO M COORDINATES
C   INPUT-   XD,YD,ZD DIPOLE COORDINATES, ZD NORTH, XD TAILWARD
C   OUTPUT-  XM,YM,ZM M COORDINATES, XM IS ANTISUNWARD
C   VARIABLES- SNPS AND CSPS, SIN AND COS OF DIPOLE TILT ANGLE
C
C   COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
C
C   XM = ZD*SNPS + XD*CSPS
C   YM = YD
```

MAP25530
MAP25540
MAP25550
MAP25560
MAP25570
MAP25580
MAP25590
MAP25600
MAP25610
MAP25620
MAP25630
MAP25640
MAP25650
MAP25660
MAP25670
MAP25680
MAP25690
MAP25700
MAP25710
MAP25720
MAP25730
MAP25740
MAP25750
MAP25760
MAP25770
MAP25780
MAP25790
MAP25800
MAP25810
MAP25820
MAP25830
MAP25840
MAP25850
MAP25860
MAP25870
MAP25880
MAP25890
MAP25900
MAP25910
MAP25920
MAP25930
MAP25940
MAP25950
MAP25960
MAP25970
MAP25980
MAP25990
MAP26000
MAP26010
MAP26020
MAP26030
MAP26040
MAP26050
MAP26060
MAP26070
MAP26080
MAP26090
MAP26100
MAP26110
MAP26120
MAP26130
MAP26140
MAP26150
MAP26160

90/06/30
12:04:14

40

mapint.for

```
      ZM = ZD*CSPS - XD*SNPS
C
      RETURN
      END
C *****:      SPHRCO
      SUBROUTINE SPHRCO (NDIM, KSA)
C
C  PURPOSE-  CALCULATE COEFFICIENTS FOR LEGENDRE POLYNOMIALS
C  INPUT-    NDIM, NUMBER OF GRID POINTS FOR NUMERICAL INTEGRATION
C            GRID POINTS ARE ON SPHERE SURFACE FROM THETA 0 TO
C            180 AND PHI=0, THETA=0 POINTS DOWN TAIL
C            KSA, ITERATION LEVEL
C  OUTPUT-   G1,G0, COEFFICIENTS FOR TILT 0 AND 90 DEGREES RESPECT.
C            CONTAIN ALL EXCEPT ANGULAR AND LEGENDRE FUNCTIONS
C            SEE EQ. 5.2.3 REF4 FOR GENERAL EXPRESSIONS
C  VARIABLES- Q, RATIO OF DIPOLE MOMENT/RADIUS**3
C            R, RADIUS OF TAIL AND SPHERE
C            B, = R - STANDOFF DISTANCE
C            N, ORDER OF LEGENDRE POLYNOMIAL
C            TI, DIPOLE TILT ANGLE
C            HS, GRID SIZE
C            Y1,Y0, INTEGRAND FOR 0 AND 90 DEGREE CASES
C            Z1,Z0, RESULT OF INTEGRALS FOR 0 AND 90 DEGREE CASES
C
      COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
      COMMON /CHABLO/ G1(20), G0(20)
      COMMON /BLOCK1/ B,R,Q,PI,PSI
C
C
      IF (KSA.EQ.1) THEN
C          COEFFICIENTS ARE ANALYTIC, 5.11 AND 5.12 REF1
          TI = 0.0
          CALL CONST(TI)
          G1(1) = - 2. * Q
          G0(1) = - 2. * Q
          EP = +1.0
          RRP = 1.0
          RP = B/R
C
          DO 1 N = 2, 20
              EP = -EP
              RRP = RRP * RP
              G1(N) = - Q * EP * FLOAT(N+1) * RRP
              G0(N) = - Q * EP * FLOAT(N*(N+1)) * RRP
1          CONTINUE
      ELSE
C          NUMERICAL INTEGRATION REQUIRED FOR KSA>1
C          GET RADIAL COMPONENT OF DIPOLE PLUS TAIL SOLUTION
C          ON GRID
          CALL BRDIP (NDIM)
          DO 3 N = 1, 20
              FORM INTEGRANDS Y1 AND Y0
              CALL FUN1 (N,NDIM)
              INTEGRATE, RESULTS Z1 AND Z0
              CALL INTEGR (HS,Y1,Z1,NDIM)
              CALL INTEGR (HS,Y0,Z0,NDIM)
              ZFE = FLOAT(2*N+1)
              FN = FLOAT(N)
C              FORM COEFFICIENTS 5.2.3 REF4
              G1(N) = 0.5 * ZFE / (FN*(FN+1.)) * Z1(NDIM)
              G0(N) = 0.5 * ZFE * Z0(NDIM)
3          CONTINUE
      END IF
C
```

90/06/30
12:04:14

mapint.for

41

```

      RETURN
      END
C *****: DAYTOT
      SUBROUTINE DAYTOT(RR,T,PHI,HR,HT,HP)
C
C PURPOSE- CALCULATE TOTAL FIELD OF DAYSIDE HALF OF SPHERE ZV<0
C INPUT- RR,T,PHI SPHERICAL COORDINATES OF INTEREST
C OUTPUT- HR,HT,HP FIELD COMPONENTS IN SPHERICAL SYSTEM
C VARIABLES- CDIPOL, FRAC. OF DIPOLE PENETRATING OUT OF MAGNETOPAUSE
C             CIMF, FRAC. OF IMF PENETRATING INTO MAGNETOPAUSE
C
      COMMON /BLOCK1/ B, R, Q, PI, PSI
      COMMON /FIMF/ HIMF, CDIPOL, CIMF, XI
C
      CD = (1.- CDIPOL)
      CF = (1.- CIMF)
C
      IF(RR.LE.R) THEN
C          GET FIELD INSIDE SPHERE
C          DIPOLE FIELD
          CALL BDIP (RR,HR1,HT1,HP1)
          CHAPMAN-FERRARO DUE TO DIPOLE
          CALL CFDSPH(RR,HR2,HT2,HP2)
          TAIL SOLUTION, DAYSIDE INFLUENCE
          CALL BTASPH (RR,HR3,HT3,HP3)
          IMF FIELD
          CALL BIMF (HIMF,XI,D1,D2,D3,D4, HR5,HT5,HP5, D5,D6,D7)
C
          EQ. 3.22 INTERNAL FIELD REF1
          HR = HR1 + CD * (HR2 + HR3) + CIMF * HR5
          HT = HT1 + CD * (HT2 + HT3) + CIMF * HT5
          HP = HP1 + CD * (HP2 + HP3) + CIMF * HP5
      ELSE
C          GET FIELD OUTSIDE SPHERE
C          DIPOLE FIELD
          CALL BDIP (RR,HR1,HT1,HP1)
          CHAPMAN-FERRARO DUE TO IMF
          CALL CFISPH(RR,HR2,HT2,HP2)
          IMF FIELD
          CALL BIMF (HIMF,XI,D1,D2,D3,D4, HR5,HT5,HP5, D5,D6,D7)
C
          EQ. 3.23 EXTERNAL FIELD REF1
          HR = HR5 + CF * HR2 + CDIPOL * HR1
          HT = HT5 + CF * HT2 + CDIPOL * HT1
          HP = HP5 + CF * HP2 + CDIPOL * HP1
      END IF
      RETURN
      END
C *****: TAILTO
      SUBROUTINE TAILTO(RR,PHI,Z,HR,HP,HZ)
C
C PURPOSE- CALCULATE TOTAL FIELD OF CYLINDER PORTION FOR ZV>0
C INPUT- RR,PHI,Z, CYLINDRICAL COORDINATES
C OUTPUT- HR,HP,HZ, FIELD COMPONENTS IN CYLINDRICAL SYSTEM
C VARIABLES- CDIPOL, FRAC. OF DIPOLE PENETRATING OUT OF MAGNETOPAUSE
C             CIMF, FRAC. OF IMF PENETRATING IN THROUGH MAGNETOPAUSE
C             SNPH,CSPH, SIN AND COS OF PHI
C
      COMMON /BLOCK1/ BEX,R,Q,PI,PSI
      COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT, ST
      COMMON /FIMF/ HIMF, CDIPOL, CIMF, XI
C
      CD = (1.- CDIPOL)
```

mapint for

CF = (1.- CIMF)

IF (RR.LE.R) THEN

GET FIELD INSIDE CYLINDER

RRR = SQRT(RR*RR + Z*Z)

DIPOLE FIELD

CALL BDIP(RRR,HR1,HT1,HP1)

TRANSFORM SPHERICAL TO CARTESIAN, HZV DOWN TAIL

CALL BTRAN1(HR1,HT1,HP1, HXV,HYV,HZV)

TRANSFORM CARTEIAN TO CYLINDRICAL

HR1 = HXV * CSPH + HYV * SNPH

HP1 = HYV * CSPH - HXV * SNPH

HZ1 = HZV

TOTAL TAIL FIELD SOLUTION

CALL BTATA(Z,RR,HR2,HZ2,HP2)

CHAPMAN-FERRARO = TOTAL FIELD - DIPOLE FIELD

HR2 = HR2 - HR1

HP2 = HP2 - HP1

HZ2 = HZ2 - HZ1

GET IMF FIELD

CALL BIMF (HIMF,XI, D1,D2,D3,D4,D5,D6,D7, HR5,HZ5,HP5)

SUM OF INTERNAL FIELDS, 3.22 REF1

HR = HR1 + CD * HR2 + CIMF * HR5

HZ = HZ1 + CD * HZ2 + CIMF * HZ5

HP = HP1 + CD * HP2 + CIMF * HP5

ELSE

GET FIELD OUTSIDE OF CYLINDER

RRR = SQRT(RR*RR + Z*Z)

DIPOLE FIELD

CALL BDIP(RRR,HR1,HT1,HP1)

TRANSFORM SPHERICAL TO CARTESIAN, HZV DOWN TAIL

CALL BTRAN1(HR1,HT1,HP1, HXV,HYV,HZV)

TRANSFORM CARTESIAN TO CYLINDRICAL

HR1 = HXV * CSPH + HYV * SNPH

HP1 = HYV * CSPH - HXV * SNPH

HZ1 = HZV

CHAPMAN-FERRARO DUE TO IMF

CALL CFITA(RR,HR2,HZ2,HP2)

IMF FIELD

CALL BIMF (HIMF,XI, D1,D2,D3,D4,D5,D6,D7, HR5,HZ5,HP5)

SUM OF EXTERNAL FIELDS USING 3.23 REF1

HR = HR5 + CF * HR2 + CDIPOL * HR1

HZ = HZ5 + CF * HZ2 + CDIPOL * HZ1

HP = HP5 + CF * HP2 + CDIPOL * HP1

END IF

RETURN

END

*****: BTRAN1

SUBROUTINE BTRAN1(HRV,HTV,HPV,HXV,HYV,HZV)

PURPOSE- CHANGES FIELD COMPONENTS FROM SPHERICAL TO CARTESIAN

HZV POINTS DOWN TAIL AXIS, =0 TO SPHERE CENTER

INPUT- HRV,HTV,HPV, SPHERICAL COMPONENTS

OUTPUT- HXV,HYV,HZV, CARTESIAN COMPONENTS

VARIABLES- ST,CT, SIN AND COS OF THETA

CP,SP, COS AND SIN OF PHI

COMMON /BLOCK5/ CP,SP,QCS,QSN,CSPS,SNPS

COMMON /BLOCK6/ CT, ST

HXV = HRV*ST*CP + HTV*CT*CP - HPV*SP

HYV = HRV*ST*SP + HTV*CT*SP + HPV*CP

HZV = HRV*CT - HTV*ST

MAP27450

MAP27460

MAP27470

MAP27480

MAP27490

MAP27500

MAP27510

MAP27520

MAP27530

MAP27540

MAP27550

MAP27560

MAP27570

MAP27580

MAP27590

MAP27600

MAP27610

MAP27620

MAP27630

MAP27640

MAP27650

MAP27660

MAP27670

MAP27680

MAP27690

MAP27700

MAP27710

MAP27720

MAP27730

MAP27740

MAP27750

MAP27760

MAP27770

MAP27780

MAP27790

MAP27800

MAP27810

MAP27820

MAP27830

MAP27840

MAP27850

MAP27860

MAP27870

MAP27880

MAP27890

MAP27900

MAP27910

MAP27920

MAP27930

MAP27940

MAP27950

MAP27960

MAP27970

MAP27980

MAP27990

MAP28000

MAP28010

MAP28020

MAP28030

MAP28040

MAP28050

MAP28060

MAP28070

MAP28080

```
RETURN
END
C *****: TAILCO
SUBROUTINE TAILCO(NDIM)
C
C PURPOSE- NUMERICAL CALCULATION OF FOURIER-BESSEL COEFFICIENTS
C INPUT- NDIM, NUMBER OF GRID POINTS ALONG CYLINDER CAP AT
C THETA=90 DEGREES FOR INTEGRATION
C OUTPUT- AX,BX FOURIER-BESSEL COEFFICIENTS WITH MODIFICATIONS
C TO BE USEFUL IN FINAL EXPRESSION 6.4 REF1
C VARIABLES- BE,BO BESSEL FUNCTIONS J1 AND J0 WITH ARGUMENTS THAT
C ARE ROOTS OF DERIVATIVES OF THE BESSEL FUNCTIONS
C SEE 5.21 AND 5.22 REF1
C Z1,Z0 RESULTS OF INTEGRATION FOR 0 AND 90 DEGREES CASES
C ZE,Z0 ZEROS OF DERIVATIVES OF J1 AND J0
C Q, DIPOLE MOMENT/(RADIUS**3)
C
COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
COMMON /BLOCK1/ B ,R, Q, PI, PSI
COMMON /BLOCK2/ ZE(20), AX(20), ZO(20), BX(20)
COMMON /BLOCK7/ BE(20), BO(20)
C
C GENERATE POTENTIAL ON CYLINDER HEAD= CHAP-FERR + DIPOLE
CALL TAPOT(NDIM)
C GENERATE COEFFICIENTS FOR N ORDERS
DO 1 N = 1, 20
C FORM INTEGRANDS Y1 AND Y0
CALL FUN2(N,NDIM)
C DO INTEGRATIONS FOR BOTH TILT CASES,RESULT Z1, Z0
CALL INTEGR(HS,Y1,Z1,NDIM)
CALL INTEGR(HS,Y0,Z0,NDIM)
C PREPARE SOME BESSEL FACTORS
BX0 = BE(N) * BE(N)
BY0 = BO(N) * BO(N)
ZEX = ZE(N) * ZE(N)
C COEFFICIENTS A AND B OF 5.24 AND 5.25 REF1
AX(N) = Z1(NDIM) * ZEX / (BX0 * (ZEX-1.))
BX(N) = Z0(NDIM) / (BY0)
C MODIFY COEFFICIENTS TO SUIT EXPRESSION 6.4 REF1
AX(N) = AX(N) * ZE(N) * Q
BX(N) = BX(N) * ZO(N) * 2. * Q
1 CONTINUE
C
RETURN
END
C *****: TRANS3
SUBROUTINE TRANS3(XV,YV,ZV,RR,T,PHI)
C
C PURPOSE- CHANGE CARTESIAN TO SPHERICAL AND FORM TRIG. FUNCTIONS
C FOR COMMON BLOCKS
C INPUT- XV,YV,ZV CARTESIAN WITH ZV DOWN TAIL
C OUTPUT- RR,T,PHI SPHERICAL COORDINATES
C VARIABLES- Q, RATIO DIPOLE MOMENT/(RADIUS**3)
C
COMMON /BLOCK1/ BEX,R,Q,PI,PSI
COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
COMMON /BLOCK6/ CT, ST
C
RR = SQRT(XV*XV + YV*YV + ZV*ZV)
PHI = ATAN3(YV,XV)
T = ATAN3(SQRT(XV*XV+YV*YV),ZV)
C
CT = COS(T)
ST = SIN(T)
```


90/06/30
12:04:14

mapint.for

44

CSPH = COS(PHI)
SNPH = SIN(PHI)
QCS = Q*CSPH
QSN = Q*SNPH

C

RETURN
END

C *****: TRANS5
SUBROUTINE TRANS5(XM,YM,ZM, XV,YV,ZV, XGSE,YGSE,ZGSE)

C

C PURPOSE- TRANSFORM M TO SPHERICAL AND GSE SYSTEM
C INPUT- XM,YM,ZM THE M SYSTEM, EARTH CENTERED, XM DOWN TAIL
C OUTPUT- XV,YV,ZV CARTESIAN, SPHERE CENTERED, ZV DOWN TAIL
C XGSE,YGSE,ZGSE, GSE SYSTEM
C VARIABLES- BEX, = RADIUS - STAND-OFF DISTANCE

C

COMMON /BLOCK1/ BEX,R,Q,PI,PSI
XV = ZM
YV = -YM
ZV = XM - BEX

C

XGSE = - XM
YGSE = -YM
ZGSE =+ ZM

C

RETURN
END

C *****: ZERO
SUBROUTINE ZERO

C

C PURPOSE- FILL COMMON BLOCKS WITH ZEROS OF BESSELS AND FUNCTION
C VALUES AT ZEROS
C INPUT- SOURCE IS TABLE 9.5 REF2
C ZE,ZEROS OF J1'(XI)
C ZO, ZEROS OF JO'(YI)
C BE, BESSEL VALUE J1(XI)
C BO, BESEL VALUE JO(YI)

C

COMMON /BLOCK2/ ZE(20),AX(20),ZO(20),BX(20)
COMMON /BLOCK7/ BE(20),BO(20)

C

C

C

ZEROS J1'(XI) = 0.0

C

ZE(1)= 1.84118
ZE(2)= 5.33144
ZE(3)= 8.53632
ZE(4)= 11.70600
ZE(5)= 14.86359
ZE(6)= 18.01553
ZE(7)= 21.16437
ZE(8)= 24.31133
ZE(9)= 27.45705
ZE(10)= 30.60192
ZE(11)= 33.74618
ZE(12)= 36.88999
ZE(13)= 40.03344
ZE(14)= 43.17663
ZE(15)= 46.31960
ZE(16)= 49.46239
ZE(17)= 52.60504
ZE(18)= 55.74757
ZE(19)= 58.89000
ZE(20)= 62.03235

MAP28730
MAP28740
MAP28750
MAP28760
MAP28770
MAP28780
MAP28790
MAP28800
MAP28810
MAP28820
MAP28830
MAP28840
MAP28850
MAP28860
MAP28870
MAP28880
MAP28890
MAP28900
MAP28910
MAP28920
MAP28930
MAP28940
MAP28950
MAP28960
MAP28970
MAP28980
MAP28990
MAP29000
MAP29010
MAP29020
MAP29030
MAP29040
MAP29050
MAP29060
MAP29070
MAP29080
MAP29090
MAP29100
MAP29110
MAP29120
MAP29130
MAP29140
MAP29150
MAP29160
MAP29170
MAP29180
MAP29190
MAP29200
MAP29210
MAP29220
MAP29230
MAP29240
MAP29250
MAP29260
MAP29270
MAP29280
MAP29290
MAP29300
MAP29310
MAP29320
MAP29330
MAP29340
MAP29350
MAP29360

90/06/30
12:04:14

mapint.for

45

C
C
C
C

ZEROS $J_0'(YI) = 0.0$

ZO(1) = 3.8317059702
ZO(2) = 7.0155866698
ZO(3) = 10.1734681351
ZO(4) = 13.3236919363
ZO(5) = 16.4706300509
ZO(6) = 19.6158585105
ZO(7) = 22.7600843806
ZO(8) = 25.9036720876
ZO(9) = 29.0468285349
ZO(10) = 32.1896799110
ZO(11) = 35.3323075501
ZO(12) = 38.4747662348
ZO(13) = 41.6170942128
ZO(14) = 44.7593189977
ZO(15) = 47.9014608872
ZO(16) = 51.0435351836
ZO(17) = 54.1855536411
ZO(18) = 57.3275254379
ZO(19) = 60.4694578453
ZO(20) = 63.61141

C
C
C
C

BE(I) = J1(XI) BESSELS EVALUATED AT XI'S OF ABOVE

BE(1) = +0.58187
BE(2) = -0.34613
BE(3) = +0.27330
BE(4) = -0.23330
BE(5) = +0.20701
BE(6) = -0.18802
BE(7) = +0.17346
BE(8) = -0.16184
BE(9) = +0.15228
BE(10) = -0.14424
BE(11) = +0.13736
BE(12) = -0.13137
BE(13) = +0.12611
BE(14) = -0.12143
BE(15) = +0.11724
BE(16) = -0.11345
BE(17) = +0.11001
BE(18) = -0.10687
BE(19) = +0.10397
BE(20) = -0.10131

C
C
C
C

BO(I) = J0(YI) BESSELS EVALUATED AT YI'S OF ABOVE

BO(1) = -0.4027593957
BO(2) = +0.3001157525
BO(3) = -0.2497048871
BO(4) = +0.2183594072
BO(5) = -0.1964653715
BO(6) = +0.1800633753
BO(7) = -0.1671846005
BO(8) = +0.1567249836
BO(9) = -0.1480111100
BO(10) = +0.1406057982
BO(11) = -0.1342112403
BO(12) = +0.1286166221

MAP29370
MAP29380
MAP29390
MAP29400
MAP29410
MAP29420
MAP29430
MAP29440
MAP29450
MAP29460
MAP29470
MAP29480
MAP29490
MAP29500
MAP29510
MAP29520
MAP29530
MAP29540
MAP29550
MAP29560
MAP29570
MAP29580
MAP29590
MAP29600
MAP29610
MAP29620
MAP29630
MAP29640
MAP29650
MAP29660
MAP29670
MAP29680
MAP29690
MAP29700
MAP29710
MAP29720
MAP29730
MAP29740
MAP29750
MAP29760
MAP29770
MAP29780
MAP29790
MAP29800
MAP29810
MAP29820
MAP29830
MAP29840
MAP29850
MAP29860
MAP29870
MAP29880
MAP29890
MAP29900
MAP29910
MAP29920
MAP29930
MAP29940
MAP29950
MAP29960
MAP29970
MAP29980
MAP29990
MAP30000

90/06/30
12:04:14

mapint.for

46

```
BO(13) = -0.1236679608
BO(14) = +0.1192498120
BO(15) = -0.1152736941
BO(16) = +0.1116704969
BO(17) = -0.1083853489
BO(18) = +0.1053740554
BO(19) = -0.1026005671
BO(20) = +0.1000

C
C
RETURN
END
C *****: BIMF
SUBROUTINE BIMF(HIMF,XI, HXM,HZM, HXV,HZV, HR,HT,HP, HRHO,HZ,HPhi)
C
C PURPOSE- DETERMINE IMF CONTRIBUTION IN CARTESIAN M AND V, SPHER,
C AND CYLINDRICAL COORDINATES
C NO Y COMPONENT INCLUDED
C INPUT- HIMF, MAGNITUDE OF IMF (GAMMA)
C XI, DIRECTION OF IMF, 0 DEGREES IS NORTHWARD, 90 ANTISUN
C OUTPUT- HXM,HZM FIELD VALUES IN M SYSTEM, HXM DOWN TAIL
C HXV,HZV FIELD VALUES IN V SYSTEM, HZV DOWN TAIL
C HR,HT,HP " " " SPHERICAL
C HRHO,HZ,HPhi " " CYLINDRICAL
C VARIABLES- ST,CT SIN AND COS OF THETA
C SNPH,CSPH SIN AND COS OF PHI
C
COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
COMMON /BLOCK6/ CT, ST
C DEGREES TO RADIANS
AXI = XI*3.1415926535/180.0
C M SYSTEM
HXM = HIMF * SIN(AXI)
HZM = HIMF * COS(AXI)
C V SYSTEM
HXV = HZM
HZV = HXM
C SPHERICAL SYSTEM
HR = + HXV*ST*CSPH + HZV*CT
HT = + HXV*CT*CSPH - HZV*ST
HP = - HXV*SNPH
C CYLINDRICAL SYSTEM
HRHO = + HXV * CSPH
HPhi = + HP
HZ = + HZV
C
RETURN
END
C *****: CFDSPH
SUBROUTINE CFDSPH(RR,HR,HT,HP)
C
C PURPOSE- CALCULATE FIELD OF CHAPMAN-FERRARO CURRENTS FOR
C SHIELDING DIPOLE ONLY IN SPHERE. SUMMATION OF FIELD DUE
C TO POTENTIAL 5.11 REF1
C NOTE: RECURRENCE RELATIONS JAHNKE-EMDE NORMALIZED, REF3
C INPUT- RR, RADIAL LOCATION
C OUTPUT- HR,HT,HP SPHERICAL FIELD COMPONENTS, 5.2.13 REF4
C VARIABLES- GAM1,GAM0 LEGENDRE POLYNOMIAL COEFFICIENTS FROM SPHRCO
C PN, LEGENDRE POLYNOMIAL M=0
C CPN, DERIVATIVE OF PN W.R.T. ARGUMENT COS(THETA)
C TPN, DERIVATIVE OF PN W.R.T. THETA
C PNM, LEGENDRE POLYNOMIAL M=1
C TPNM, DERIVATIVE OF PNM W.R.T. THETA
C SPNM, QUOTIENT PN/SIN(THETA)
```

90/06/30
12:04:14

mapint.for

47

```
C          31,C1 SIN AND COS OF THETA
C
C          DIMENSION PN(20), CPN(20), TPN(20)
C          DIMENSION PNM(20),TPNM(20),SPNM(20)
C          COMMON /CHABLO/ GAM1(20), GAM0(20)
C
C          COMMON /BLOCK1/ B,R,Q,PI,PSI
C          COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
C          COMMON /BLOCK6/ C1, S1
C
C          D = 1.0E-03
C
C          C2 = C1*C1 - S1*S1
C          S2 = 2.*S1*C1
C
C          PN(1) = C1
C          PN(2) = 0.25*(3.*C2+1.)
C          CPN(1) = 1.0
C          CPN(2) = 3.*C1
C
C          DO 1 N=3,20
C              FN = FLOAT(N)
C              ZFE = 2.*FN - 1.0
C              PN(N) = (1./FN) * (ZFE*C1*PN(N-1) - (FN-1.)*PN(N-2))
C              CPN(N) = ZFE*PN(N-1) + CPN(N-2)
1      CONTINUE
C
C          DO 2 N=1,20
C              FN = FLOAT(N)
C              TPN(N) = - S1*CPN(N)
C              PNM(N) = - TPN(N)
C              SPNM(N) = + CPN(N)
C              TPNM(N) = FN*(FN+1.)*PN(N) - C1*CPN(N)
2      CONTINUE
C
C          HR1 = GAM1(1) * PNM(1)
C          HR2 = GAM0(1) * PN(1)
C          HT1 = GAM1(1) * TPNM(1)
C          HT2 = GAM0(1) * TPN(1)
C          HP = GAM1(1) * SPNM(1)
C
C          RP = 0.0
C          IF(RR.GT.D) RP = (RR/R)
C          RRP = 1.0
C
C          DO 3 N=2,20
C              FN = FLOAT(N)
C
C              RRP = RRP * RP
C
C              GAMA = GAM1(N) * RRP
C              GAMB = GAM0(N) * RRP
C
C              1 AND 2 INDICATE TERMS MEANT FOR 0 AND 90 DEGREE CASES
C              HR1 = HR1 + GAMA * PNM(N)
C              HR2 = HR2 + GAMB * PN(N)
C
C              HT1 = HT1 + GAMA * TPNM(N) / FN
C              HT2 = HT2 + GAMB * TPN(N) / FN
C
C              HP = HP + GAMA * SPNM(N) / FN
C
3      CONTINUE
C
```

MAP30650
MAP30660
MAP30670
MAP30680
MAP30690
MAP30700
MAP30710
MAP30720
MAP30730
MAP30740
MAP30750
MAP30760
MAP30770
MAP30780
MAP30790
MAP30800
MAP30810
MAP30820
MAP30830
MAP30840
MAP30850
MAP30860
MAP30870
MAP30880
MAP30890
MAP30900
MAP30910
MAP30920
MAP30930
MAP30940
MAP30950
MAP30960
MAP30970
MAP30980
MAP30990
MAP31000
MAP31010
MAP31020
MAP31030
MAP31040
MAP31050
MAP31060
MAP31070
MAP31080
MAP31090
MAP31100
MAP31110
MAP31120
MAP31130
MAP31140
MAP31150
MAP31160
MAP31170
MAP31180
MAP31190
MAP31200
MAP31210
MAP31220
MAP31230
MAP31240
MAP31250
MAP31260
MAP31270
MAP31280

90/06/30
12:04:14

48

mapint.for

```
C      EACH PART RELATING TO TILT CASES IS COMBINED WITH
C      FUNCTIONS OF 5.9 REF1
      HR = - (HR1 * CSPH * CSPS + HR2 * SNPS)
      HT = - (HT1 * CSPH * CSPS + HT2 * SNPS)
      HP = + HP * SNPH * CSPS
C
C      RETURN
C      END
C *****: CFISPH
      SUBROUTINE CFISPH(RR,HR,HT,HP)
C
C      PURPOSE- CALCULATE CHAPMAN-FERRARO FIELD OUTSIDE SHERE DUE TO IMF
C                USING 5.15 REF1
C      INPUT-    RR, RADIAL COORDINATE
C      OUTPUT-   HR,HT,HP CHAP-FERR FIELD COMPONENTS IN SPHERICAL SYSTEM
C      VARIABLES- HR3,HT3,HP3 IMF FIELD COMPONENTS FROM SUBROUT. BIMF
C
C      COMMON /BLOCK1/ BEX, R, Q, PI, PSI
C      COMMON /FIMF/ HIMF, CDIPOL, CIMF, XI
C
C      GET IMF FIELD COMPONENTS
C      CALL BIMF(HIMF,XI, D1,D2,D3,D4, HR3,HT3,HP3, D5,D6,D7)
C      TRANSFORM COMPONENTS TO SPHERICAL SYSTEM
      FR = R/RR
      FR = FR*FR*FR
C
      HR = - HR3 * FR
      HT = + 0.5 * HT3 * FR
      HP = + 0.5 * HP3 * FR
C
      RETURN
      END
C *****: BDIP
      SUBROUTINE BDIP(RR,HR,HT,HP)
C
C      PURPOSE- CALCULATE EXPLICIT COMPONENTS OF DIPOLE USING SPHERICAL
C                EQUIVALENT OF 5.6 REF1 FOR POTENTIAL
C      INPUT-    RR, RADIAL COORDINATE LOCATION
C      OUTPUT-   HR,HT,HP DIPOLE FIELD IN SPHERICAL SYSTEM (GAMMA)
C      VARIABLES- C1,S1 COS AND SIN OF THETA
C                R, RADIUS OF SPHERE AND TAIL
C                B, = R - STANDOFF DISTANCE
C                SNPS,CSPS SIN AND COS OF PSI, THE DIPOLE TILT ANGLE
C
C      COMMON /BLOCK1/ B,R,Q,PI,PSI
C      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
C      COMMON /BLOCK6/ C1, S1
C
C      R3 = R**3
C
C      A = QCS*R3
C      AA = QSN*R3
C
C      V = ((RR*RR) + (B*B) + (2.*B*RR*C1))
C
C      AV = ABS(V)
C      SV = SIGN(1.0,V)
C
C      W3 = SQRT(AV**3)
C      W4 = SQRT(AV**5)
C
C      W1 = W3 * SV
```

MAP31290
MAP31300
MAP31310
MAP31320
MAP31330
MAP31340
MAP31350
MAP31360
MAP31370
MAP31380
MAP31390
MAP31400
MAP31410
MAP31420
MAP31430
MAP31440
MAP31450
MAP31460
MAP31470
MAP31480
MAP31490
MAP31500
MAP31510
MAP31520
MAP31530
MAP31540
MAP31550
MAP31560
MAP31570
MAP31580
MAP31590
MAP31600
MAP31610
MAP31620
MAP31630
MAP31640
MAP31650
MAP31660
MAP31670
MAP31680
MAP31690
MAP31700
MAP31710
MAP31720
MAP31730
MAP31740
MAP31750
MAP31760
MAP31770
MAP31780
MAP31790
MAP31800
MAP31810
MAP31820
MAP31830
MAP31840
MAP31850
MAP31860
MAP31870
MAP31880
MAP31890
MAP31900
MAP31910
MAP31920

90/06/30
12:04:14

mapint.for

49

```
C      W2 = W4 * SV
C
C      F1 = 2.*RR*B*S1
C      G1 = 2.*RR + 2.*B*C1
C      E1 = 1.5*RR*G1/W2
C      E2 = 1.5*F1/W2
C
C      HR1 = A*CSPS*S1*(1./W1 - E1)
C      HR2 = Q*R3*C1*SNPS*(1./W1 - E1)
C      HR3 = -Q*R3*B*SNPS*3.*(RR+ B*C1)/W2
C
C      HT1 = A*CSPS*(C1/W1 + S1*E2)
C      HT2 = Q*R3*SNPS*(C1*E2 - S1/W1)
C      HT3 = Q*R3*B*SNPS*(3.*B*S1/W2)
C
C      HP = -AA*CSPS/W1
C
C      HR = HR1 + HR2 + HR3
C      HT = HT1 + HT2 + HT3
C      HP = HP
C
C      RETURN
C      END
C *****:      FUN1
C      SUBROUTINE FUN1(N,NDIM)
C
C      PURPOSE-   FORM INTEGRAND FOR COEFFICIENT CALCULATIONS AT EACH OF
C                  NDIM STEPS FOR A CERTAIN ORDER OF LEGENDRE POLYNOMIAL
C      INPUT-      N, ORDER OF LEGENDRE POLYNOMIAL
C                  NDIM, NUMBER OF GRID POINTS USED FOR INTEGRATION
C      OUTPUT-     Y1,Y0 VALUE OF INTEGRAND FOR CASES 0 AND 90 DEGREES RESPECTIVELY
C      VARIABLES-  PNM, LEGENDRE POLYNOMIAL M=1
C                  PN,  LEGENDRE POLYNOMIAL M=0
C                  ST, SIN(T) LAST VALUE COMES FROM SUBROUT. LEGEN
C
C      COMMON /LEG1/ PN(20), PNM(20), CPN(20)
C      COMMON /BLOCK8/ W1(1001),W0(1001)
C      COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
C      COMMON /BLOCK6/ CT, ST
C
C      DO 1 I = 1, NDIM
C          X(I) IS THETA, GENERATES LEGENDRE POLYNOMIAL TERMS
C          CALL LEGEN(X(I),N)
C
C          D = 1.0E-20
C
C          IF (ABS(W1(I)).LT.D) W1(I) = 0.0
C          IF (ABS(W0(I)).LT.D) W0(I) = 0.0
C          IF (ABS(PNM(N)).LT.D) PNM(N) = 0.0
C          IF (ABS(PN(N)).LT.D) PN(N) = 0.0
C          IF (ABS(ST).LT.D) ST = 0.0
C
C          NDIM INTEGRANDS FOR EACH ORDER TO GET COEFFICIENTS
C          5.2.3 REF4
C          Y1(I) = W1(I) * PNM(N) * ST
C          Y0(I) = W0(I) * PN(N) * ST
C
C      1 CONTINUE
C
C      RETURN
C      END
C *****:      BRDIP
```

MAP31930
MAP31940
MAP31950
MAP31960
MAP31970
MAP31980
MAP31990
MAP32000
MAP32010
MAP32020
MAP32030
MAP32040
MAP32050
MAP32060
MAP32070
MAP32080
MAP32090
MAP32100
MAP32110
MAP32120
MAP32130
MAP32140
MAP32150
MAP32160
MAP32170
MAP32180
MAP32190
MAP32200
MAP32210
MAP32220
MAP32230
MAP32240
MAP32250
MAP32260
MAP32270
MAP32280
MAP32290
MAP32300
MAP32310
MAP32320
MAP32330
MAP32340
MAP32350
MAP32360
MAP32370
MAP32380
MAP32390
MAP32400
MAP32410
MAP32420
MAP32430
MAP32440
MAP32450
MAP32460
MAP32470
MAP32480
MAP32490
MAP32500
MAP32510
MAP32520
MAP32530
MAP32540
MAP32550
MAP32560

90/06/30
12:04:14

mapint.for

50

```

SUBROUTINE BRDIP (NDIM)
C
C PURPOSE- GENERATE RADIAL COMPONENTS OF DIPOLE FIELD ON GRID FOR
C TILT 0 AND 90 DEGREES
C INPUT- NDIM, NUMBER OF GRID POINTS AT RADIUS=R ALONG CURVE
C THETA=0 TO 180 DEGREES TO BE USED FOR INTEGRATION
C OUTPUT- W1,W0 RADIAL COMPONENTS OF DIPOLE, INTO BLOCK 8
C VARIABLES- KDIM, NUMBER OF THE GRID POINT SEPARATING DAY AND NIGHT-
C SIDE OF SPHERE
C R, RADIUS OF SHPERE AND TAIL
C Q, RATIO DIPOLE MOMENT/(R**3)
C SNPH,CSPH SIN AND COS OF PHI
C QSN,QCS =Q*SNPH AND =Q*CSPH RESPECTIVELY
C X(I), THETA VALUES OF GRID POINTS
C HS, GRID POINT SPACING
C
COMMON /BLOCK8/ W1(1001),W0(1001)
COMMON /BLOCK9/ X(1001), Z1(1001),Y0(1001),Z1(1001),Z0(1001), HS
COMMON /BLOCK1/ B,R,Q,PI,PSI
COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
COMMON /BLOCK6/ CT,ST
COMMON /BLOCK3/ H0,ALFA,KSUIT

C
C KDIM = (NDIM-1)/2
C
C PHI = 0 DEGREES FOR THIS SUBROUT.
C
CSPH = 1.0
C SNPH = 0.0
C QCS = Q
C QSN = 0.0
C
C
C GENERATE GRID POINTS
C
C X(1) = 0.0
C HS = PI/FLOAT(NDIM-1)
C DO 1 I = 2, NDIM
C X(I) = FLOAT(I-1) * HS
1 CONTINUE
C
C GET COMPONENTS FOR TILT=0 DEGREES
C
C TI = 0.0
C CALL CONST(TI)
C
C DO 4 I = 1, NDIM
C
C CT = COS(X(I))
C ST = SIN(X(I))
C
C RRDIP = R
C GET RADIAL COMPONENT OF DIPOLE AT SPHERE RADIUS
C CALL BDIP(RRDIP,HR,HT,HP)
C W1(I) = HR
C
C IF (I.LE.KDIM) THEN
C LOCATION IS NIGHTSIDE, CHANGE TO CYLINDRICAL
C RHO= R * ST
C Z = R * CT
C GET CYLINDER SOLUTION
C CALL VACTA(Z,RHO,HR,HZ,HP)
C HR = HR * ST + HZ * CT
C CYLINDER SOLUTION HAS DIPOLE + CHAP-FERR 5.16 REF1
C W1(I) = W1(I) - HR
C ELSE
C LOCATION IS DAYSIDE OF SPHERE
```

90/06/30
12:04:14

mapint.for

51

```
C      RRSHE = R
C      DAYSIDE FIELD RESULTING FROM TAIL SOLUTION
C      CALL BTASPH(RRSHE,HR,HT,HP)
C      DIPOLE PLUS DAYSIDE TAIL SOLUTION GIVES DAYSIDE
C      FIELD, 5.1 REF1
C      W1(I) = W1(I) + HR
C      END IF
4     CONTINUE
C
C      GENERATE COMPONENTS FOR TILT=90 DEGREES
C      (SAME PROCEDURE AS ABOVE FOR TILT=0 DEGREES)
C      TI = 90.0
C      CALL CONST(TI)
C
C      DO 7 I = 1, NDIM
C
C          CT = COS(X(I))
C          ST = SIN(X(I))
C
C          RRDIP = R
C          CALL BDIP(RRDIP,HR,HT,HP)
C          W0(I) = HR
C
C          IF (I.LE.KDIM) THEN
C              RHO= R * ST
C              Z = R * CT
C              CALL VACTA(Z,RHO,HR,HZ,HP)
C              HR = HR * ST + HZ * CT
C              W0(I) = W0(I) - HR
C          ELSE
C              RRSHE = R
C              CALL BTASPH(RRSHE,HR,HT,HP)
C              W0(I) = W0(I) + HR
C          END IF
7     CONTINUE
C
C      RETURN
C      END
C *****:      INTEGR
C      SUBROUTINE INTEGR(H,Y,Z,NDIM)
C
C      PURPOSE-    GENERAL INTEGRATION ROUTINE
C      INPUT-      H, GRID SIZE
C      Y, INTEGRAND, FROM FUN1 OR FUN2 SUBROUT.
C      NDIM, NUMBER OF GRID POINTS USED
C      OUTPUT-     Z INTEGRAL RESULT OF FUNCTION Y FROM ZERO TO NDIM*H
C      (USED TO FILL Z1 AND Z0 OF BLOCK 9)
C
C      DIMENSION Y(NDIM), Z(NDIM)
C
C      HT=.3333333*H
C      IF (NDIM-5) 7,8,1
C
C      NDIM IS GREATER THAN 5. PREPARATIONS OF INTEGRATION LOOP
1     SUM1=Y(2)+Y(2)
C      SUM1=SUM1+SUM1
C      SUM1=HT*(Y(1)+SUM1+Y(3))
C      AUX1=Y(4)+Y(4)
C      AUX1=AUX1+AUX1
C      AUX1=SUM1+HT*(Y(3)+AUX1+Y(5))
C      AUX2=HT*(Y(1)+3.875*(Y(2)+Y(5))+2.625*(Y(3)+Y(4))+Y(6))
C
C      MAP33210
C      MAP33220
C      MAP33230
C      MAP33240
C      MAP33250
C      MAP33260
C      MAP33270
C      MAP33280
C      MAP33290
C      MAP33300
C      MAP33310
C      MAP33320
C      MAP33330
C      MAP33340
C      MAP33350
C      MAP33360
C      MAP33370
C      MAP33380
C      MAP33390
C      MAP33400
C      MAP33410
C      MAP33420
C      MAP33430
C      MAP33440
C      MAP33450
C      MAP33460
C      MAP33470
C      MAP33480
C      MAP33490
C      MAP33500
C      MAP33510
C      MAP33520
C      MAP33530
C      MAP33540
C      MAP33550
C      MAP33560
C      MAP33570
C      MAP33580
C      MAP33590
C      MAP33600
C      MAP33610
C      MAP33620
C      MAP33630
C      MAP33640
C      MAP33650
C      MAP33660
C      MAP33670
C      MAP33680
C      MAP33690
C      MAP33700
C      MAP33710
C      MAP33720
C      MAP33730
C      MAP33740
C      MAP33750
C      MAP33760
C      MAP33770
C      MAP33780
C      MAP33790
C      MAP33800
C      MAP33810
C      MAP33820
C      MAP33830
C      MAP33840
```



```
SUM2=Y(5)+Y(5)
SUM2=SUM2+SUM2
SUM2=AUX2-HT*(Y(4)+SUM2+Y(6))
Z(1)=0.
AUX=Y(3)+Y(3)
AUX=AUX+AUX
Z(2)=SUM2-HT*(Y(2)+AUX+Y(4))
Z(3)=SUM1
Z(4)=SUM2
IF (NDIM-6) 5,5,2
C
C   INTEGRATION LOOP
2 DO 4 I=7,NDIM,2
  SUM1=AUX1
  SUM2=AUX2
  AUX1=Y(I-1)+Y(I-1)
  AUX1=AUX1+AUX1
  AUX1=SUM1+HT*(Y(I-2)+AUX1+Y(I))
  Z(I-2)=SUM1
  IF (I-NDIM) 3,6,6
3  AUX2=Y(I)+Y(I)
  AUX2=AUX2+AUX2
  AUX2=SUM2+HT*(Y(I-1)+AUX2+Y(I+1))
4  Z(I-1)=SUM2
5  Z(NDIM-1)=AUX1
  Z(NDIM)=AUX2
  RETURN
6  Z(NDIM-1)=SUM2
  Z(NDIM)=AUX1
  RETURN
C   END OF INTEGRATION LOOP
C
7 IF (NDIM-3) 12,11,8
C
C   NDIM IS EQUAL TO 4 OR 5
8 SUM2=1.125*HT*(Y(1)+Y(2)+Y(2)+Y(2)+Y(3)+Y(3)+Y(3)+Y(4))
  SUM1=Y(2)+Y(2)
  SUM1=SUM1+SUM1
  SUM1=HT*(Y(1)+SUM1+Y(3))
  Z(1)=0.
  AUX1=Y(3)+Y(3)
  AUX1=AUX1+AUX1
  Z(2)=SUM2-HT*(Y(2)+AUX1+Y(4))
  IF (NDIM-5) 10,9,9
9  AUX1=Y(4)+Y(4)
  AUX1=AUX1+AUX1
  Z(5)=SUM1+HT*(Y(3)+AUX1+Y(5))
10 Z(3)=SUM1
  Z(4)=SUM2
  RETURN
C
C   NDIM IS EQUAL TO 3
11 SUM1=HT*(1.25*Y(1)+Y(2)+Y(2)-.25*Y(3))
  SUM2=Y(2)+Y(2)
  SUM2=SUM2+SUM2
  Z(3)=HT*(Y(1)+SUM2+Y(3))
  Z(1)=0.
  Z(2)=SUM1
12 RETURN
END
C *****:      BTRAN2
  SUBROUTINE BTRAN2 (HRV,HZV,HPV,HXV,HYV,HZVV)
C
C   PURPOSE-  CHANGE FIELD COMPONENTS FROM CYLINDRICAL TO CARTESIAN
C
```

MAP33850
MAP33860
MAP33870
MAP33880
MAP33890
MAP33900
MAP33910
MAP33920
MAP33930
MAP33940
MAP33950
MAP33960
MAP33970
MAP33980
MAP33990
MAP34000
MAP34010
MAP34020
MAP34030
MAP34040
MAP34050
MAP34060
MAP34070
MAP34080
MAP34090
MAP34100
MAP34110
MAP34120
MAP34130
MAP34140
MAP34150
MAP34160
MAP34170
MAP34180
MAP34190
MAP34200
MAP34210
MAP34220
MAP34230
MAP34240
MAP34250
MAP34260
MAP34270
MAP34280
MAP34290
MAP34300
MAP34310
MAP34320
MAP34330
MAP34340
MAP34350
MAP34360
MAP34370
MAP34380
MAP34390
MAP34400
MAP34410
MAP34420
MAP34430
MAP34440
MAP34450
MAP34460
MAP34470
MAP34480

90/06/30
12:04:14

53

mapint.for

```
C INPUT- HRV,HZV,HPV CYLINDRICAL FIELD COMPONENTS, HZV DOWN TAILMAP34490
C OUTPUT- HXV,HYV,HZVV CARTESIAN COMPONENTS, HZVV DOWN TAIL MAP34500
C VARIABLES- SP,CP SIN AND COS OF PHI MAP34510
C MAP34520
C COMMON /BLOCK5/ CP,SP,QCS,QSN,CSPS,SNPS MAP34530
C MAP34540
C HXV = HRV*CP - HPV*SP MAP34550
C HYV = HRV*SP + HPV*CP MAP34560
C HZVV = HZV MAP34570
C MAP34580
C RETURN MAP34590
C END MAP34600
C *****: BTASPH MAP34610
C SUBROUTINE BTASPH(RR,HR,HT,HP) MAP34620
C MAP34630
C PURPOSE- CALCULATE FIELD COMPONENTS OF TAILFIELD WITHIN SPHERE MAP34640
C FOR ZV<0. (IMAGE OF VACTA) COMPONENTS DERIVED FROM 5.26 MAP34650
C REF1 WITH SIGN CHANGE IN EXPONENT DUE TO ZV POSITION MAP34660
C FIRST CALLED FROM SPHRCO ON ITERATION 2. 15 TERMS USED. MAP34670
C INPUT- RR, RADIAL DISTANCE IN SPHERICAL COORDINATES MAP34680
C OUTPUT- HR,HT,HP SPHERICAL FIELD COMPONENTS MAP34690
C VARIABLES- RHO,Z CYLINDRICAL EQUIVALENTS OF INPUT MAP34700
C ST,CT SIN AND COS OF THETA MAP34710
C ZE, ZEROS OF J1'( ) BESSEL 5.23 REF1 MAP34720
C ZO, ZEROS OF JO'( ) BESSEL " MAP34730
C AX,BX FOURIER-BESSEL COEFFICIENTS WITH ADDITIONS MAP34740
C ALPHA, PLASMA CONTENT OF TAIL 1 MEANS VACUUM MAP34750
C 0 MEANS HARRIS SHEET MAP34760
C ALPHA=1 USED TO GET COEFFICIENTS MAP34770
C ALPHA NOT = 1 ONLY POSSIBLE FOR BTOTAL APPLICATION MAP34780
C MAP34790
C MAP34800
C COMMON /BLOCK1/ B,R,Q,PI,PSI MAP34810
C COMMON /BLOCK2/ ZE(20),AX(20),ZO(20),BX(20) MAP34820
C COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS MAP34830
C COMMON /BLOCK6/ CT,ST MAP34840
C COMMON /BLOCK3/ H0,ALFA,KSWIT MAP34850
C MAP34860
C CHANGE INPUT TO CYLINDRICAL COORDINATES MAP34870
C RHO = RR * ST MAP34880
C Z = RR * CT MAP34890
C MAP34900
C FL = 0.5 * (1.-ALFA) MAP34910
C MAP34920
C HHR = 0.0 MAP34930
C HHZ = 0.0 MAP34940
C HHP = 0.0 MAP34950
C THR = 0.0 MAP34960
C THZ = 0.0 MAP34970
C D = 1.0E-02 MAP34980
C MAP34990
C MAP35000
C DO 100 I = 1,15 MAP35010
C MAP35020
C ARX = ZE(I)/R * RHO MAP35030
C ARZ = ZE(I)/R * Z MAP35040
C TARX = ZO(I)/R * RHO MAP35050
C TARZ = ZO(I)/R * Z MAP35060
C MAP35070
C EX = EXP(+ARZ) MAP35080
C EX = EX * FL * AX(I) MAP35090
C TEX = EXP(+TARZ) MAP35100
C TEX = TEX * FL * BX(I) MAP35110
C MAP35120
```

```
      B0 = 1.0
      B1 = 0.0
      B2 = 0.0
      TB0 = 1.0
      TB1 = 0.0
      IF (ABS(RHO).GE.1.0E-06) THEN
        B0 = BFJ0(ARX)
        B1 = BFJ1(ARX)
        B2 = BFJ2(B1,B0,ARX)
        TB0 = BFJ0(TARX)
        TB1 = BFJ1(TARX)
      END IF
      HHR = HHR + (B0-B2)*EX
      HHP = HHP + (B0+B2)*EX
      HHZ = HHZ + 2.0*B1*EX
C
      THR = THR + TB1*TEX
      THZ = THZ + TB0*TEX
C
C
C 100 CONTINUE
C
      HHR = HHR*CSPS*CSPH
      HHP = HHP*CSPS*SNPH
      HHZ = HHZ*CSPS*CSPH
      THR = THR*SNPS
      THZ = THZ*SNPS
C
C
      HHR = - HHR + THR
      HHP = + HHP
      HHZ = - HHZ - THZ
C      OUTPUT IN SPHERICAL COORDINATES
      HR = HHR*ST + HHZ*CT
      HT = HHR*CT - HHZ*ST
      HP = HHP
C
      RETURN
      END
C *****: ATAN3
      FUNCTION ATAN3(Y,X)
C
      PI = 3.1415926535
      ATAN3 = 0.00
      IF ((X.NE.0.0).OR.(Y.NE.0.0)) ATAN3 = ATAN2(Y,X)
C
      RETURN
      END
C *****: BFJ0
      FUNCTION BFJ0(X)
C
C PURPOSE- CALCULATE BESSEL FUNCTION J0(X)
C          USING 9.4.1 AND 9.4.3 OF REF2 TO ORDER 12 APPROXIMATION
C INPUT-   X, ARGUMENT OF THE BESSEL FUNCTION
C OUTPUT-  BFJ0(X), =J0(X)
C
      DATA A1/-2.2499997/,A2/1.2656208/,A3/-0.3163866/,A4/0.0444479/,
1 A5/-0.0039444/,A6/0.00021/,B0/0.79788456/,B1/-0.00000077/,
2 B2/-0.0055274/,B3/-0.00009512/,B4/0.00137237/,B5/-0.00072805/,
3 B6/0.00014476/,C0/-0.78539816/,C1/-0.04166397/,C2/-0.00003954/,
4 C3/0.00262573/,C4/-0.00054125/,C5/-0.00029333/,C6/0.00013558/
C
      X0=ABS(X)
      T=X0/3.0
```

```
MAP35130
MAP35140
MAP35150
MAP35160
MAP35170
MAP35180
MAP35190
MAP35200
MAP35210
MAP35220
MAP35230
MAP35240
MAP35250
MAP35260
MAP35270
MAP35280
MAP35290
MAP35300
MAP35310
MAP35320
MAP35330
MAP35340
MAP35350
MAP35360
MAP35370
MAP35380
MAP35390
MAP35400
MAP35410
MAP35420
MAP35430
MAP35440
MAP35450
MAP35460
MAP35470
MAP35480
MAP35490
MAP35500
MAP35510
MAP35520
MAP35530
MAP35540
MAP35550
MAP35560
MAP35570
MAP35580
MAP35590
MAP35600
MAP35610
MAP35620
MAP35630
MAP35640
MAP35650
MAP35660
MAP35670
MAP35680
MAP35690
MAP35700
MAP35710
MAP35720
MAP35730
MAP35740
MAP35750
MAP35760
```

90/06/30
12:04:14

mapint.for

55

```
C
      IF (T.LT.1.0) THEN
C          USE EQ. 9.4.1
            T=T*T
            H1=1.0+T*(A1+T*(A2+T*(A3+T*(A4+T*(A5+T*A6))))
            BFJ0=H1
      ELSE
C          USE EQ. 9.4.3
            T=1.0/T
            H2=B0+T*(B1+T*(B2+T*(B3+T*(B4+T*(B5+T*B6))))
            H3=X0+C0+T*(C1+T*(C2+T*(C3+T*(C4+T*(C5+T*C6))))
            BFJ0=H2*COS(H3)/SQRT(X0)
      END IF
      RETURN
      END
C *****:      BFJ1
      FUNCTION BFJ1(X)
C
C      PURPOSE-    CALCULATE BESSEL FUNCTION J1(X)
C                  USING 9.4.4 AND 9.4.6 REF2 TO ORDER 12 APPROXIMATION
C      INPUT-      X, ARGUMENT OF BESSEL FUNCTION
C      OUTPUT-     BFJ1, BESSEL FUNCTION J1(X)
C
      DATA A1/-.56249985/,
1  A2/.21093573/,A3/-.03954289/,A4/.00443319/,A5/-.00031761/,
2  A6/.00001109/,B0/.79788456/,B1/.00000156/,B2/.01659667/,
3  B3/.00017105/,B4/-.00249511/,B5/.00113653/,B6/-.00020033/,
4  C0/-2.35619449/,C1/.12499612/,C2/.0000565/,C3/-.00637879/,
5  C4/.00074348/,C5/.00079824/,C6/-.00029166/
C
      X0=ABS(X)
      X1=SIGN(1.0,X)
      T=X0/3.0
C
      IF (T.LT.1.0) THEN
C          USE 9.4.4
            T=T*T
            H1=0.5+T*(A1+T*(A2+T*(A3+T*(A4+T*(A5+T*A6))))
            BFJ1=X*H1
      ELSE
C          USE 9.4.6
            T=1.0/T
            H2=B0+T*(B1+T*(B2+T*(B3+T*(B4+T*(B5+T*B6))))
            H3=X0+C0+T*(C1+T*(C2+T*(C3+T*(C4+T*(C5+T*C6))))
            BFJ1=X1*H2*COS(H3)/SQRT(X0)
      END IF
      RETURN
      END
C *****:      BFJ2
      FUNCTION BFJ2(B1,B0,X)
C
C      PURPOSE-    CALCULATE BESSEL FUNCTION J2(X) USING 9.1.27 REF2
C      INPUT-      B1, J1(X) BESSEL FROM FUNCTION BFJ1
C                  B0, J0(X) BESSEL FROM FUNCTION BFJ0
C                  X, ARGUMENT OF BESSEL FUNCTIONS
C      OUTPUT-     BFJ2, BESSEL J2(X)
C
      BFJ2 = 0.0
      IF(X.GT.0.0) THEN
          BFJ2 = (2./X) * B1 - B0
      END IF
C
      RETURN
      END
```

MAP35770
MAP35780
MAP35790
MAP35800
MAP35810
MAP35820
MAP35830
MAP35840
MAP35850
MAP35860
MAP35870
MAP35880
MAP35890
MAP35900
MAP35910
MAP35920
MAP35930
MAP35940
MAP35950
MAP35960
MAP35970
MAP35980
MAP35990
MAP36000
MAP36010
MAP36020
MAP36030
MAP36040
MAP36050
MAP36060
MAP36070
MAP36080
MAP36090
MAP36100
MAP36110
MAP36120
MAP36130
MAP36140
MAP36150
MAP36160
MAP36170
MAP36180
MAP36190
MAP36200
MAP36210
MAP36220
MAP36230
MAP36240
MAP36250
MAP36260
MAP36270
MAP36280
MAP36290
MAP36300
MAP36310
MAP36320
MAP36330
MAP36340
MAP36350
MAP36360
MAP36370
MAP36380
MAP36390
MAP36400

90/06/30
12:04:14

56

mapint.for

```
C *****: CFITA MAP36410
SUBROUTINE CFITA(RR,HR,HZ,HP) MAP36420
C MAP36430
C PURPOSE- CALCULATE CHAPMAN-FERRARO FIELD OF IMF OUTSIDE CYLINDER MAP36440
C INPUT- RR RHO IN CYLINDRICAL COORDINATES MAP36450
C OUTPUT- HR,HZ,HP CYLINDRICAL FIELD COMPONENTS MAP36460
C VARIABLES- HIMF, STRENGTH OF IMF FIELD MAP36470
C XI, IMF ANGLE MAP36480
C R, RADIUS OF MAGNETOPAUSE SPHERE AND TAIL MAP36490
C MAP36500
COMMON /BLOCK1/ BEX, R, Q, PI, PSI MAP36510
COMMON /FIMF/ HIMF, CDIPOL, CIMF, XI MAP36520
C MAP36530
C GET IMF FIELD COMPONENTS MAP36540
CALL BIMF(HIMF,XI, D1,D2,D3,D4,D5,D6,D7, HR3,HZ3,HP3) MAP36550
C MAP36560
FR = R/RR MAP36570
FR = FR*FR*FR MAP36580
C MAP36590
C COMPONENTS DERIVED FROM 5.30 REF1 MAP36600
HR = - HR3 * FR MAP36610
HZ = + 0.5 * HZ3 * FR MAP36620
HP = + 0.5 * HP3 * FR MAP36630
C MAP36640
RETURN MAP36650
END MAP36660
C *****: FUN2 MAP36670
SUBROUTINE FUN2(N,NDIM) MAP36680
C MAP36690
C PURPOSE- CALCULATE INTEGRAND FOR FOURIER-BESSEL INTEGRALS. ONE MAP36700
C FOR EACH OF NDIM GRID POINTS. DONE FOR TILT 0 AND 90 DEG MAP36710
C INPUT- N, ORDER OF BESSEL FUNCTION MAP36720
C NDIM, NUMBER OF GRID POINTS MAP36730
C OUTPUT- Y1,Y0 INTEGRANDS FOR TILT 0, 90 DEGREES, 5.24,5.25 REF1 MAP36740
C OUTPUT TO BLOCK 9 MAP36750
C VARIABLES- ZE, ZEROS OF DERIVATIVES OF BESSEL J1 MAP36760
C ZO, " " " " " J0 MAP36770
C X, RADIAL VALUES OF GRID POINTS MAP36780
C R, RADIUS OF TAIL CYLINDER MAP36790
C MAP36800
COMMON /BLOCK8/ W1(1001),W0(1001) MAP36810
COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS MAP36820
COMMON /BLOCK1/ B ,R, Q, PI, PSI MAP36830
COMMON /BLOCK2/ ZE(20), AX(20), ZO(20), BX(20) MAP36840
C MAP36850
C MAP36860
DO 1 I = 1, NDIM MAP36870
C ARGUMENTS OF BESSELS MAP36880
ARX = ZE(N) * X(I) / R MAP36890
TARX = ZO(N) * X(I) / R MAP36900
C SET BESSEL VALUES FOR VERY SMALL ARGUMENTS MAP36910
B1 = 0.0 MAP36920
B0 = 1.0 MAP36930
IF(ABS(X(I)).GE.1.0E-06) THEN MAP36940
C GET BESSEL FUNCTION VALUES J1 AND J0 MAP36950
B1 = BFJ1(ARX) MAP36960
B0 = BFJ0(TARX) MAP36970
END IF MAP36980
C INTEGRANDS OF 5.24 AND 5.25 REF1 MAP36990
Y1(I) = X(I) * W1(I) * B1 MAP37000
Y0(I) = X(I) * W0(I) * B0 MAP37010
1 CONTINUE MAP37020
C MAP37030
C MAP37040
```

```
      RETURN
      END
C *****:      LEGEN
      SUBROUTINE LEGEN(T,NE)
C
C  PURPOSE-  CALCULATE LEGENDRE POLYNOMIAL TERMS
C  INPUT-    T, THETA OF SPHERICAL SYSTEM
C            NE, ORDER OF LEGENDRE POLYNOMIAL
C  OUTPUT-   PN, LEGENDRE POLYNOMIAL TERMS M=0
C            CPN, DERIVATIVE OF PN W.R.T. COS(T)
C            PNM, LEGENDRE POLYNOMIAL TERMS M=1
C
      COMMON /LEG1/ PN(20), PNM(20), CPN(20)
      COMMON /BLOCK6/ C1, S1
C
      C1 = COS(T)
      S1 = SIN(T)
C
      C2 = C1*C1 - S1*S1
      S2 = 2.*S1*C1
C
      PN(1) = C1
      PN(2) = 0.25*(3.*C2+1.)
C
      CPN(1) = 1.0
      CPN(2) = 3.*C1
C
      PNM(1) = + S1 * CPN(1)
      PNM(2) = + S1 * CPN(2)
C
      IF (NE.LT.3)      RETURN
      IF (NE.EQ.3)      NELE= 4
      IF (NE.GT.3)      NELE= NE
C
      DO 1  N=3,NELE
          FN = FLOAT(N)
          ZFE = 2.*FN - 1.0
C          RECURRENCE RELATION FROM REF3 JAHNKE-EMDE NORMALIZED
          PN(N) = (1./FN) * (ZFE*C1*PN(N-1) - (FN-1.)*PN(N-2))
          CPN(N) = ZFE*PN(N-1) + CPN(N-2)
          PNM(N) = + S1*CPN(N)
1      CONTINUE
C
      RETURN
      END
C *****:      BTATA
      SUBROUTINE BTATA(Z,RHO,HHR,HHZ,HHP)
C
C  PURPOSE-  CALCULATE COMPONENTS OF TAILFIELD SOLUTION FOR ZV>0
C            INCLUDING STRETCH,SUMMARIES 6.3 AND 6.4 REF1
C  INPUT-    Z,RHO CYLINDRICAL COORDINATES
C  OUTPUT-   HHR,HHZ,HHP CYLINDRICAL FIELD COMPONENTS
C  VARIABLES- ALFA, PLASMA CONTENT STRETCH PARAMETER
C            SNPH,CSPH SIN AND COS OF PHI, A CYLINDRICAL COORDINATE
C            SNPS,CSPS SIN AND COS OF PSI, THE DIPOLE TILT ANGLE
C            AX,BX COEFFICIENTS FOR FOURIER-BESSEL SERIES
C            R, RADIUS OF TAIL
C            ZE,ZO ZEROS OF BESSEL FUNCTION DERIVATIVES 5.25 REF1
C
      COMMON /BLOCK1/ B,R,Q,PI,PSI
      COMMON /BLOCK2/ ZE(20),AX(20),ZO(20),BX(20)
      COMMON /BLOCK5/CSPH,SNPH,QCS,QSN,CSPS,SNPS
```

MAP37050
MAP37060
MAP37070
MAP37080
MAP37090
MAP37100
MAP37110
MAP37120
MAP37130
MAP37140
MAP37150
MAP37160
MAP37170
MAP37180
MAP37190
MAP37200
MAP37210
MAP37220
MAP37230
MAP37240
MAP37250
MAP37260
MAP37270
MAP37280
MAP37290
MAP37300
MAP37310
MAP37320
MAP37330
MAP37340
MAP37350
MAP37360
MAP37370
MAP37380
MAP37390
MAP37400
MAP37410
MAP37420
MAP37430
MAP37440
MAP37450
MAP37460
MAP37470
MAP37480
MAP37490
MAP37500
MAP37510
MAP37520
MAP37530
MAP37540
MAP37550
MAP37560
MAP37570
MAP37580
MAP37590
MAP37600
MAP37610
MAP37620
MAP37630
MAP37640
MAP37650
MAP37660
MAP37670
MAP37680

900630
12-04-14

mapint.for

58

COMMON /BLOCK6/ CT,ST
COMMON /BLOCK3/ H0,ALFA,KSWIT

HHR = 0.0
HHZ = 0.0
HHP = 0.0
THR = 0.0
THZ = 0.0
D = 1.0E-02

IF (ABS (ALFA) .LT. 1.0E-06) THEN
ALFA = 1.0E-06
END IF

FL = 0.5 * (1.-ALFA)

DO 100 I = 1,15

ARGUMENTS OF BESSEL FUNCTIONS

ARX = ZE(I)/R*RHO
TARX = ZO(I)/R*RHO

ARZ = ZE(I)/R*Z
TARZ = ZO(I)/R*Z

FLZX = ALFA * ARZ
FLZY = ALFA * TARZ

EX = 0.0
TEX = 0.0
IF ((ARZ.LT.170.) .AND. (TARZ.LT.170.)) THEN
EX = EXP (-ARZ)
TEX = EXP (-TARZ)

EX = EX * FL * AX(I)
TEX = TEX * FL * BX(I)

END IF

FLEX = 0.0
FLEY = 0.0
IF ((FLZX.LT.170.) .AND. (FLZY.LT.170.)) THEN
FLEX = EXP (-FLZX)
FLEY = EXP (-FLZY)
END IF

FLEX = FLEX * ALFA * AX(I)
FLEY = FLEY * ALFA * BX(I)

B0 = 1.0
B1 = 0.0
B2 = 0.0
TB0 = 1.0
TB1 = 0.0

IF (ABS (RHO) .GE. 1.0E-06) THEN
GET BESSEL FUNCTION
B0 = BFJ0 (ARX)
B1 = BFJ1 (ARX)
B2 = BFJ2 (B1,B0,ARX)
TB0 = BFJ0 (TARX)
TB1 = BFJ1 (TARX)

END IF

HHR = HHR + (B0-B2) * (FLEX + EX)

MAP37690
MAP37700
MAP37710
MAP37720
MAP37730
MAP37740
MAP37750
MAP37760
MAP37770
MAP37780
MAP37790
MAP37800
MAP37810
MAP37820
MAP37830
MAP37840
MAP37850
MAP37860
MAP37870
MAP37880
MAP37890
MAP37900
MAP37910
MAP37920
MAP37930
MAP37940
MAP37950
MAP37960
MAP37970
MAP37980
MAP37990
MAP38000
MAP38010
MAP38020
MAP38030
MAP38040
MAP38050
MAP38060
MAP38070
MAP38080
MAP38090
MAP38100
MAP38110
MAP38120
MAP38130
MAP38140
MAP38150
MAP38160
MAP38170
MAP38180
MAP38190
MAP38200
MAP38210
MAP38220
MAP38230
MAP38240
MAP38250
MAP38260
MAP38270
MAP38280
MAP38290
MAP38300
MAP38310
MAP38320

9005/30
12:04:14

mapint.for

59

```
C      HHP = HHP + (B0+B2) * (FLEX + EX)
      HHZ = HHZ + 2.*B1 * (FLEX/ALFA + EX)
      THR = THR + TB1 * (FLEY + TEX)
      THZ = THZ + TB0 * (FLEY/ALFA + TEX)
C
100  CONTINUE
C      SUPERPOSITION OF 0 AND 90 DEGREE TILT CASES
      HHR = HHR*CSPS*CSPH
      HHP = HHP*CSPS*SNPH
      HHZ = HHZ*CSPS*CSPH
      THR = THR*SNPS
      THZ = THZ*SNPS
C
C      HHR = +HHR -THR
      HHP = -HHP
      HHZ = -HHZ -THZ
C
      RETURN
      END
C *****: VACTA
      SUBROUTINE VACTA(Z,RR,HHR,HHZ,HHP)
C
C  PURPOSE-  CALCULATE NIGHTSIDE (ZV>0) VACUUM FIELD COMPONENTS OF
C             TAIL SOLUTION FROM POTENTIAL 5.26 REF1. SOLUTION
C             INCLUDES DIPOLE PLUS CHAPMAN-FERRARO. FIRST CALLED ON
C             SECOND ITERATION AS FIRST ITER. HAS ANALYTIC COEFFICIENTS
C  INPUT-    Z,RR Z AND RHO OF CYLINDRICAL SYSTEM
C  OUTPUT-   HHR,HHZ,HHP FIELD COMPONENTS IN CYLINDRICAL SYSTEM
C  VARIABLES- ZE,ZO ZEROS OF DERIVATIVES OF BESSELS J1 AND J0
C             AX,BX COEFFICIENTS OF FOURIER-BESSEL SERIES FROM TAILCOM
C             5.24 AND 5.25 REF1
C             R, RADIUS OF TAIL
C             SNPS,CSPS SIN AND COS OF PSI, THE DIPOLE TILT ANGLE
C             SNPH,SNPH SIN AND COS OF PHI, A CYLINDRICAL COORDINATE
C
      COMMON /BLOCK1/ B,R,Q,PI,PSI
      COMMON /BLOCK2/ ZE(20),AX(20),ZO(20),BX(20)
      COMMON /BLOCK5/ CSPH,SNPH,QCS,QSN,CSPS,SNPS
      COMMON /BLOCK6/ CT,ST
C
      HHR = 0.0
      HHZ = 0.0
      HHP = 0.0
      THR = 0.0
      THZ = 0.0
      D = 1.0E-02
C
C      DO 100 I = 1,15
C          ARGUMENTS FOR BESSEL FUNCTIONS
          ARX = ZE(I)/R*RR
          ARZ = ZE(I)/R*Z
          TARX = ZO(I)/R*RR
          TARZ = ZO(I)/R*Z
C
          EX = 0.0
          TEX = 0.0
C
          IF ((ARZ.LT.170.).AND.(TARZ.LT.170.)) THEN
              EX = EXP(-ARZ)
              EX = EX*AX(I)
```


90/06/30
12:04:14

mapint.for

60

```

      TEX = EXP(-TARZ)
      TEX = TEX*BX(I)
      END IF
C
      B0 = 1.0
      B1 = 0.0
      B2 = 0.0
      TB0 = 1.0
      TB1 = 0.0
C
      IF (ABS(RR).GE.1.0E-06) THEN
        B0 = BFJ0(ARX)
        B1 = BFJ1(ARX)
        B2 = BFJ2(B1,B0,ARX)
        TB0 = BFJ0(TARX)
        TB1 = BFJ1(TARX)
      END IF
C
      HHR = HHR + (B0-B2)*EX
      HHP = HHP + (B0+B2)*EX
      HHZ = HHZ + 2.0*B1*EX
C
      THR = THR + TB1*TEX
      THZ = THZ + TB0*TEX
C
100  CONTINUE
C      SUPERIMPOSE 0 AND 90 DEGREE TILT CASES
      HHR = HHR*CSPS*CSPH
      HHP = HHP*CSPS*SNPH
      HHZ = HHZ*CSPS*CSPH
      THR = THR*SNPS
      THZ = THZ*SNPS
C
C
      HHR = +HHR -THR
      HHP = -HHP
      HHZ = -HHZ -THZ
C
      RETURN
      END
C *****: TAPOT
      SUBROUTINE TAPOT(NDIM)
C
C      PURPOSE- GET SUM OF CHAPMAN-FERRARO AND DIPOLE POTENTIALS AT NDIM
C                LOCATIONS ON THE CYLINDER CAP ZV=0
C      INPUT- NDIM, NUMBER OF GRID POINTS FOR INTEGRATION
C      OUTPUT- W1,W0 SUM OF POTENTIALS FOR TILTS 0 AND 90 DEGREE
C                OUTPUT IS POTENTIAL*(1/DIPOLE MOMENT0
C      VARIABLES- HS, GRID SIZE
C                  R, RADIUS OF TAIL
C                  X, RADIAL POSITIONS OF GRID POINTS
C
      COMMON /BLOCK8/ W1(1001),W0(1001)
      COMMON /BLOCK9/ X(1001),Y1(1001),Y0(1001),Z1(1001),Z0(1001), HS
      COMMON /BLOCK1/ B, R, Q, PI, PSI
C
C      GENERATE GRID POINTS ON CYLINDER CAP
      X(1) = 0.0
      HS = R/FLOAT(NDIM-1)
      DO 1 I = 2, NDIM
        X(I) = FLOAT(I-1) * HS
1    CONTINUE
C
C      GET POTENTIALS OF CHAPMAN-FERRARO TILT 0 AND 90 DEGREES
```

90/06/30
12:04:14

mapint.for

61

```
C      EQ. 5.18-5.20 REF1
DO 2 I = 1, NDM
  CALL CFPOT(X(I),POT1,POT0)
  W1(I) = POT1
  W0(I) = POT0
2  CONTINUE
C
C      GET POTENTIALS OF DIPOLE,TILT 0 AND 90 DEGREES 5.17 REF1
DO 3 I = 1, NDM
  CALL DIPOT(X(I),POT1,POT0)
  W1(I) = W1(I) + POT1
  W0(I) = W0(I) + POT0
3  CONTINUE
C
C      RETURN
END
C *****: CFPOT
SUBROUTINE CFPOT(RR,POT1,POT0)
C
C  PURPOSE- CALCULATE CHAPMAN-FERRARO POTENTIAL FOR A POINT AT THE
C            CYLINDER LID, ZV=0. USES 5.11 WITH THETA=90 DEGREES,
C            EQUIVALENT TO 5.18-5.20 REF1. BOTH TILT CASES 0 AND 90
C            DEGREES DONE.
C  INPUT-    RR, RHO IN CYLINDRICAL, DISTANCE FROM TAIL AXIS
C  OUTPUT-   POT1, POTENTIAL FOR 0 DEGREE TILT CASE
C            POT0, POTENTIAL FOR 90 DEGREE TILT CASE
C            OUTPUT IS ACTUALLY POTENTIAL*(-1/DIPOLE MOMENT)
C  VARIABLES- QR, DIPOLE MOMENT
C             P1,PO LEGENDRE POLYNOMIAL TERMS FOR M=1 AND 0, RESPECT.
C             G1,GO COEFFICIENTS OF LEGENDRE TERMS, M=1 AND 0 RESPECT.
C             R, RADIUS OF TAIL
C
COMMON /BLOCK1/ B, R, Q, PI, PSI
COMMON /CHABLO/ G1(20), G0(20)
C
DIMENSION P0(21), P1(21)
C
C      LEGENDRE TERMS WITH THETA=90 DEGREES
P0(2) = - 0.5
P1(1) = + 1.0
P1(3) = 3.*P0(2)
C
C
DO 1 NY = 2, 10
  EVERY OTHER TERM IS ZERO, RECURRENCE RELATION USED
  SUBROUT. LEGEN NOT CALLED
  ZFNY = 2.*FLOAT(NY)
  FAKNY = (ZFNY-1.)/ZFNY
  P0(2*NY) = - P0(2*NY-2) * FAKNY
  P1(2*NY+1) = + (ZFNY+1.) * P0(2*NY)
1  CONTINUE
C
C      TILT = 0 DEGREES CASE
RRP = RR
RP = (RR*RR)/(R*R)
C
POT1 = G1(1) * P1(1) * RR
C
DO 2 NY = 1, 9
  ZFE = FLOAT(2*NY+1)
  RRP = RRP * RP
  POT1 = POT1 + G1(2*NY+1) * P1(2*NY+1) * RRP / ZFE
2  CONTINUE
```

MAP39610
MAP39620
MAP39630
MAP39640
MAP39650
MAP39660
MAP39670
MAP39680
MAP39690
MAP39700
MAP39710
MAP39720
MAP39730
MAP39740
MAP39750
MAP39760
MAP39770
MAP39780
MAP39790
MAP39800
MAP39810
MAP39820
MAP39830
MAP39840
MAP39850
MAP39860
MAP39870
MAP39880
MAP39890
MAP39900
MAP39910
MAP39920
MAP39930
MAP39940
MAP39950
MAP39960
MAP39970
MAP39980
MAP39990
MAP40000
MAP40010
MAP40020
MAP40030
MAP40040
MAP40050
MAP40060
MAP40070
MAP40080
MAP40090
MAP40100
MAP40110
MAP40120
MAP40130
MAP40140
MAP40150
MAP40160
MAP40170
MAP40180
MAP40190
MAP40200
MAP40210
MAP40220
MAP40230
MAP40240

90/06/30
12:04:14

mapint.for

62

```
C
  QR = Q * (R**3)
  POT1 = -POT1/QR

C
C      TILT = 90 DEGREE CASE
  RRP = R
  RP  = (RR*RR)/(R*R)

C
  POT0 = 0.0

C
  DO 3 NY = 1, 10
    ZFE = FLOAT(2*NY)
    RRP = RRP * RP
    POT0 = POT0 + G0(2*NY) * P0(2*NY) * RRP / ZFE
3  CONTINUE

C
  QR = Q * (R**3)
  POT0 = -POT0/QR

C
  RETURN
  END

C *****:  DIPOT
  SUBROUTINE DIPOT(RR,POT1,POT2)

C
C  PURPOSE-  CALCULATE DIPOLE POTENTIAL AT LID OF CYLINDER USING 5.17
C             REF1 FOR TILT 0 AND 90 DEGREE CASES
C  INPUT-    RR, RHO IN CYLINDRICAL, DISTANCE FROM TAIL AXIS
C             B, = RADIUS MINUS STAND-OFF DISTANCE
C  OUTPUT-   POT1,POT2 DIPOLE POTENTIAL FOR TILT 0 AND 90 DEGREES
C             OUTPUT IS POTENTIAL*(-1/DIPOLE MOMENT)
C
C  COMMON /BLOCK1/ B, R, Q, PI, PSI

C
  RRB = SQRT(RR*RR + B*B)
  RRB = RRB**3

C
  POT1 = + RR / RRB
  POT2 = + B / RRB

C
  RETURN
  END
```

MAP40250
MAP40260
MAP40270
MAP40280
MAP40290
MAP40300
MAP40310
MAP40320
MAP40330
MAP40340
MAP40350
MAP40360
MAP40370
MAP40380
MAP40390
MAP40400
MAP40410
MAP40420
MAP40430
MAP40440
MAP40450
MAP40460
MAP40470
MAP40480
MAP40490
MAP40500
MAP40510
MAP40520
MAP40530
MAP40540
MAP40550
MAP40560
MAP40570
MAP40580
MAP40590
MAP40600
MAP40610
MAP40620
MAP40630
MAP40640
MAP40650

Appendix E

MSM Reference List: Correlation Studies

Compiled by R. V. Hilmer. Reference numbers correspond to the numbers on associated source chart. Note that "*author" means copy available

<u>No.</u>	<u>First Author</u>	<u>Source</u>
10	Akasofu, S.-I.	Magnetospheric Substorms: A Model, in Solar-Terr. Physics Part 3, ed. Dyer, 131-151, D. Reidel, 1972
20	Akasofu, S.-I.	Physics of Magnetospheric Substorms, Reidel, 1977
30	Akasofu, S.-I.	Interplanetary Energy Flux Associated with Magnetospheric Substorms, PSS, 27, 425, 1979
40	Akasofu, S.-I.	Relationships Between the AE and Dst Indices During Geomagnetic Storms, JGR, 86, 4820-4822, 1981
50	Akasofu, S.-I.	Dependence of the Geometry of the Region of Open Field Lines on the Interplanetary Magnetic Field, PSS, 29, 803, 1981
60	Akasofu, S.-I.	Energy Coupling Between the Solar Wind and the Magnetosphere, Space Sci. Rev., 28, 121, 1981
70	Akasofu, S.-I.	A High Time Resolution Study of the Solar Wind-Magnetospheric Energy Coupling Function, PSS, 30, 537, 1982
80	Akasofu, S.-I.	Dependence of the Polar Cap Geometry on the IMF, PSS, 32, 111-118, 1984
90	Aoki, T.	Influence of the Dipole Tilt Angle on the Development of Auroral Electrojets, J. Geomagn. Geoelect., 29, 441, 1977
100	Arnoldy, R. L.	Signature in the Interplanetary Medium for Substorms, JGR, 76, 5189, 1971
110	Aubry, M. P.	Magnetotail Changes in Relation to the Solar Wind Magnetic Field and Magnetospheric Substorms, JGR, 76, 4381, 1971
120	Baker, D. N.	High-Energy Magnetospheric Protons and Their Dependence on Geomagnetic and Interplanetary Conditions, JGR, 84, 7138, 1979
130	*Baker, D. N.	A High Time Resolution Study of Interplanetary Correlation with AE, GRL, 8, 179-182, 1981.
140	*Baker, D. N.	An ISEE-3 High Time Resolution Study of Interplanetary Parameter Correlations with Magnetospheric Activity, JGR, 88, 6230-6242, 1983
150	Ballif, J. R.	Further Evidence on the Correlation Between Transverse Fluctuations in the Interplanetary Magnetic Field And Kp, JGR, 74, 2289-2301, 1969
160	*Bargatze, L. F.	Magnetospheric Impulse Response for Many Levels of Geomagnetic Activity, JGR, 90, 6387-6394, 1985
170	Bobrov, M. S.	Kp Index Correlations with Solar-Wind Parameters During the First and Second Stages of a Recurrent Geomagnetic Storm, PSS, 21, 2139-2147, 1973
180	Bobrov, M. S.	Solar Wind Parameters Responsible for the Plasma Injection in the Ring Current, Astronomical Journal, 54, 1335, 1977 ???
190	Bobrov, M. S.	Variations of the Ring Current Field in Time of Being of the Earth in the High-Speed Solar Wind, Geomagn. Aeron., 21, 1048, 1981

- 200 Burch, J. L. Precipitation of Low-Energy Electrons at High-Latitudes: Effect of Interplanetary Magnetic Field and Dipole Tilt Angle, JGR, 77, 6696-6707, 1972
- 210 Burch, J. L. Rate of Erosion of Dayside Magnetic Flux Based on the Quantitative Study of the Dependence of Polar Cusp Latitude on Interplanetary Magnetic Field, Radio Science, 8, 955, 1973
- 220 Burch, J. L. Effects of Interplanetary Magnetic Sector Structure on Auroral Zone and Polar Cap Magnetic Activity, JGR, 78, 1047, 1973
- 230 Burch, J. L. Observations of Interactions Between Interplanetary and Geomagnetic Fields, Rev. Geophys. Space Phys., 12, 363, 1974
- 233 Burch, J. L. Effects of the Interplanetary Magnetic Field on the Auroral Oval and Plasmasphere, Space Sci. Review, 23, 449, 1979
- 240 Burton, R. K. An Empirical Relationship Between Dst and the Solar Wind Electric Field and Dynamic Pressure, in Proc. of Chapman Memor. Symp. of Magneto. Motions, ed. Campbell et al., 139-143, NOAA, Boulder, Colo., 1973
- 250 *Burton, R. K. An Empirical Relationship Between Interplanetary Conditions and Dst, JGR, 80, 4204-4214, 1975
- 260 Caan, M. N. Solar Wind and Substorm-Related Changes in the Lobes of the Geomagnetic Tail, JGR, 78, 8087-8096, 1973
- 270 Caan, M. N. Substorm and Interplanetary Magnetic Field Effects on the Geomagnetic Tail Lobes, JGR, 80, 191-194, 1975
- 280 Caan, M. N. Characteristics of the Association Between the Interplanetary Magnetic Field and Substorms, JGR, 82, 4837-4842, 1977
- 283 Caan, M. N. Interplanetary Magnetic Field Changes and the Magnetotail, in Dynamics of the Magnetosphere, ed. Akasofu, 63-75, D. Reidel, 1979
- 290 Candidi, M. Intensity Distribution of Dayside Polar Soft Electron Precipitation and the IMF, PSS, 31, 489-498, 1983
- 300 *Carbary, J. F. Relations Between the Interplanetary Magnetic Field Bz, AE Index, and Cusp Latitude, JGR, 91, 1549, 1986
- 310 *Carbary, J. F. Correlation of Cusp Latitude With Bz and AE(12) Using Nearly One Year's Data, JGR, 91, 10047-10054, 1986
- 320 *Carpenter, D. L. Relations Between the Dawn Minimum in the Equatorial Radius of the Plasmapause and Dst, Kp, and Local K at Byrd Station, JGR, 72, 2969-2971, 1967
- 330 *Chappell, C. R. A Study of the Influence of Magnetic Activity on the Location of the Plasmapause as Measured by OGO-5, JGR, 75, 50-56, 1970
- 340 *Chubb, T. A. Observations of the Aurora in the Far Ultraviolet from OGO 4, JGR75, 1290-1311, 1970
- 350 *Clauer, C. R. Solar Wind Control of Auroral Zone Geomagnetic Activity, GRL, 8, 915-918, 1981
- 360 Clauer, C. R. Solar Wind Control of the Low-Latitude Asymmetric Magnetic Disturbance Field, JGR, 88, 2123-2130, 1983
- 370 Crooker, N. U. On the High Correlation Between Long-Term Averages of Solar Wind Speed and Geomagnetic Activity, JGR, 82, 1933-1937, 1977
- 380 *Eather, R. H. Dayside Aurora and Relevance to Substorm Current Systems and Dayside Merging, JGR, 84, 3339, 1979

- 382 *Eather, R. H. Polar Cusp Dynamics, JGR, 90, 1569, 1985
- 388 Fairfield, D. N. Average and Unusual Locations of the Earth's Magnetopause and Bow Shock, JGR, 76, 6700, 1971
- 390 Fairfield, D. N. NASA GSFC-Preprint, X-690-73-366, 1973
- 400 *Fairfield, D. N. Solar Wind Control of Magnetospheric Pressure (CDAW 6), JGR, 90, 1201-1204, 1985
- 402 *Fairfield, D. N. Polar Rain: Solar Coronal Electrons in the Earth's Magnetosphere, JGR, 90, 4055-4068, 1985
- 410 Feldstein, Y. I. Dynamics of Auroral Belt and Polar Geomagnetic Disturbances, PSS, 15, 209, 1967
- 420 *Feldstein, Y. I. Auroras and Associated Phenomena, in Solar Terr. Physics, Part 3, ed. Dyer, 152-191, D. Reidel, 1972
- 427 Feldstein, Y. I. Electric Fields and Potential Patterns in the High-Latitude Ionosphere for Different Situations in Interplanetary Space, PSS, 32, 907-923, 1984
- 430 *Feldstein, Y. I. Ring Current Simulation in Connection with Interplanetary Space Conditions, PSS, 32, 975-984, 1984
- 440 *Feynman, J. Prediction of Geomagnetic Activity on Time Scales of One to Ten Years, Rev. Geophys. & Space Phys., 24, 650-666, 1986
- 450 Foster, J. C. Relationship of Interplanetary Parameters and Occurrence of Magnetospheric Substorms, JGR, 76, 6971, 1971
- 460 Freeman, J. W. Kp Dependence of the Plasma Sheet Boundary, JGR, 79, 4315-4317, 1974
- 470 Fukushima, N. Partial Ring Current Models for Worldwide Geomagnetic Disturbances, Rev. Geophys. & Space Phys., 11, 795-853, 1973
- 480 Garrett, H. B. The Influence of the Variance of the Interplanetary Magnetic Field on Geomagnetic Activity, Ph. D. Thesis, Rice U., Dec. 1973
- 490 *Garrett, H. B. Influence of Solar Wind Variability on Geomagnetic Activity, JGR, 79, 4603-4610, 1974
- 495 *Greenspan Simultaneous Polar Cap and Magnetotail Observations of Intense Polar Rain, JGR, 91, 11123-11132, 1986
- 500 *Gussenhoven, M., DMSP/F2 Electron Observations of Equatorward Auroral Boundaries and Their Relationship to Magnetospheric Electric Fields, JGR, 86, 768, 1981
- 502 *Gussenhoven, M., Systematics of the Equatorward Diffuse Auroral Boundary, JGR, 88, 5692-5708, 1983
- 510 Hakamada, K. Influence of the East-West Component of the Interplanetary Magnetic Field on the Intensity of the Auroral Electrojet, PSS, 28, 29, 1979
- 520 *Hardy, D. A. Response of Magnetotail Plasma at Lunar Distance to Changes in the Interplanetary Magnetic Field, Solar Wind Plasma, and Substorm Activity, JGR, 84, 1382-1390, 1979
- 530 *Hardy, D. A. DMSP/F2 Electron Observations of Equatorward Auroral Boundaries and Their Relationships to the Solar Wind Velocity and the North-South Component of the Interplanetary Magnetic Field, JGR, 86, 9961-9974, 1981
- 540 Heppner, J. P. Polar Cap Electric Field Distributions Related to the Interplanetary Magnetic Field Direction, JGR, 77, 4877, 1972

- 550 Heppner, J. P. High Latitude Electric Fields and the Modulations Related to Interplanetary Magnetic Field Parameters, *Radio Sci.*, 8, 933, 1973
- 560 *Hirshberg, J. Interplanetary Field and Geomagnetic Variations- A Unified View, *PSS*, 17, 1183-1206, 1969
- 580 *Holzer, R. E. An Evaluation of Three Predictors of Geomagnetic Activity, *JGR*, 87, 2558-2562, 1982
- 590 Holzworth, R. Mathematical Representation of the Auroral Oval, *GRL*, 2, 377, 1975
- 600 *Iyemori, T. Impulse Response of Geomagnetic Indices to Interplanetary Magnetic Field, *J. Geomag. Geoelectr.*, 31, 1-9, 1979
- 610 *Iyemori, T. Prediction of Geomagnetic Activities from Solar Wind Parameters Based on the Linear Prediction Theory, in *Solar Terr. Pred. Proc.*, 4, ed. Donnelly, US Dept. Comm., NOAA/ERL/SEL, Boulder, Colo., 1980
- 620 *Jones, D. F. A Comparison of the Geomagnetic ap Index with Fluctuations in B and V: Mariner 5, *JGR*, 79, 286-288, 1974
- 630 Kamide, Y. Dependence of the Latitude of the Cleft on the Interplanetary Magnetic Field and Substorm Activity, *JGR*, 81, 698, 1976
- 640 *Kamide, Y. Comments on No. 580 by Holzer, *JGR*, 88, 4953-4954, 1983
- 650 Kane, R. P. Relationship Between Various Indices of Geomagnetic Activity and the Interplanetary Plasma Parameters, *J. Atmos. Terr. Phys.*, 34, 1941, 1972
- 660 Kane, R. P. Relationship Between Interplanetary Plasma Parameters and Geomagnetic Dst, *JGR*, 79, 64, 1974
- 670 *Kaye, S. M. The Influence of Geomagnetic Activity on the Radial Variation of the Magnetospheric Electric Field Between L=4 and 10, *JGR*, 86, 863-867, 1981
- 680 Kendall, P. C. A Model Midnight Meridian Magnetospheric Field, *Geophys. J. R. Astron. Soc.*, 17, 185, 1969
- 690 Kivelson, M. G. The Dependence of the Polar Cusp on the North-South Component of the Interplanetary Magnetic Field, *JGR*, 78, 3761-3772, 1973
- 700 *Kivelson, M. G. Magnetospheric Electric Fields and Their Variation with Geomagnetic Activity, *Rev. Geophys. & Space Phys.*, 14, 189-197, 1976
- 710 *Kivelson, M. G. The Physics of Plasma Injection Events, in *Dynamics of the Magnetosphere*, ed. Akasofu, 385-405, D. Reidel, 1979
- 720 Maezawa, K. Dependence of the Magnetopause Position on the Southward Interplanetary Magnetic Field, *PSS*, 22, 1443, 1974
- 740 Maezawa, K. Dependence of Geomagnetic Activity on Solar Wind Parameters: A Statistical Approach, *Solar Terr. Environ. Res. Japan*, 2, 103, 1978
- 750 *Maezawa, K. Statistical Study of the Dependence of Geomagnetic Activity on Solar Wind Parameters, in *Qualitative Modeling of the Magnetosphere*, ed. Olson, 1979
- 760 *Maezawa, K. Solar Wind Velocity Effects on the Auroral Zone Magnetic Disturbances, *Sub. to Proc. Chapman Conf. SW/Magnet. Energy Coupling*, April 1985

- 768 Makita, K. The Shift of the Auroral Electron Precipitation Boundaries in the Dawn-Dusk Sector in Association with Geomagnetic Activity and the Interplanetary Magnetic Field, JGR, 88, 7967, 1983
- 770 *Makita, K. Long-Period Polar Rain Variations and the Solar-Wind, Sub. JGR, Oct. 1985
- 775 Makita, K. Temporal and Spatial Variations of the Polar Cap Dimension Inferred from the Precipitation Boundaries, JGR, 90, 2744, 1985.
- 780 *Mauk, B. H. Correlation of Kp with the Substorm-Injected Plasma Boundary, JGR, 79, 3193-3196, 1974
- 790 McDiarmid, I.B. Solar Particles and the Day side Limit of Closed Field Lines, JGR, 77, 1103, 1972
- 800 *Meloni, A. A. On the Relationships Between Interplanetary Quantities and the Global Auroral Electrojet Index, JGR, 87, 119-127, 1982
- 810 Meng, C.-I. Variations of the Magnetopause Position with Substorm Activity, JGR, 75, 3252, 1970
- 820 *Meng, C.-I. Cross-Correlation Analysis of the AE Index and the Interplanetary Magnetic Field Bz Component, JGR, 78, 617-629, 1973
- 830 Meng, C.-I. Polar Cap Variations and the Interplanetary Magnetic Field, in Dynamics of the Magnetosphere, ed. Akasofu, 23-46, D. Reidel, 1979
- 840 Meng, C.-I. Latitudinal Variations of the Polar Cusp During a Geomagnetic Storm, GRL, 9, 60, 1982
- 850 Meng, C.-I. Case Studies of the Storm Time Variation of the Polar Cusp, JGR, 88, 137, 1983
- 860 Mishin, V. M. Short-term Forecasting of Geomagnetic Storms Associated with High-Speed Solar Wind Streams, in Solar-Terr. Predictions Proceedings, 4, ed. Donnelly, A37-A45, US Depart. of Commerce, Boulder, Colo., 1980
- 870 Mozer, F. S. The Auroral Zone Electric Field, JGR, 79, 1001, 1974
- 880 Murayama, T. Effects of Solar Wind Parameters on the Development of Magnetospheric Substorms, PSS, 23, 75, 1975
- 890 *Murayama, T. Empirical Formula to Relate the Auroral Electrojet Intensity with Interplanetary Parameters, PSS, 28, 803-813, 1980
- 900 *Murayama, T. Coupling Function Between Solar Wind Parameters and Geomagnetic Indices, Rev. Geophys. & Space Phys., 20, 623-629, 1982
- 910 Murayama, T. Comparison of the Observed Values of the Dst Index with the Predicted Values by the Coupling Functions BsV, BsVV, epsilon, pBsV, Sol.-Terr. Environ. Res. Japan, 6, 39, 1982
- 920 *Murayama, T. Coupling Function Between the Solar Wind and the Dst Index, Sub. to Proc. Chapman Conf. SW/Magnet. Energy Coupling, April 1985
- 930 Parks, G. K. Correlation of Interplanetary-Space Bz Field Fluctuations and Trapped Particle Redistributions, JGR, 77, 266, 1972
- 940 Patel, V. L. Geomagnetic Activity and Size of Magnetospheric Cavity, JGR, 71, 1940, 1966

- 950 Perreault, P. A Study of Geomagnetic Substorms, *Geophys. J. R. Astro. Soc.*, 54, 547, 1978
- 960 Pike, C. P. Observed Correlations Between Interplanetary Field Variations and the Dynamics of the Auroral Oval and the High-Latitude Ionosphere, *JGR*, 79, 5129, 1974
- 970 Reiff, P. H. Dependence of Polar Cap Potential Drop on Interplanetary Parameters, *JGR*, 86, 7639, 1981
- 975 *Riehl, K. B. Average Characteristics of the Polar Rain and Their Relationship to the Solar Wind and Interplanetary Magnetic Field, *JGR*, 91, 1557-1571, 1986
- 980 *Rosenberg, R. L. A Practical Method of Predicting Geomagnetic Storms by Using Precursory Enhancements of Solar Wind Density, *JGR*, 87, 2563-2568, 1982
- 990 Rostoker, G. Relationship Between Changes in the IMF and Variations in the Magnetic Field at the Earth's Surface, *JGR*, 72, 5853-5863, 1967
- 1000 Rostoker, G. Response Time of the Magnetosphere to the Interplanetary Electric Field, *Canadian J. Phys.*, 50, 544, 1972
- 1002 Rostoker, G. Variation of Auroral Electrojet Spatial Location as a Function of the Level of Magnetospheric Activity(B), *JGR*, 91, 1716, 1986.
- 1010 *Rothwell, P. L. An Empirical Relationship Between Kp and Tail Lobe Magnetic Field, *JGR*, 91, 9012-9016, 1986
- 1020* Rudneva, N. M. Distance to the Subsolar Point of the Magnetosphere Boundary as a Function of Magnetic Activity, *Geomagn. Aeron., Engl. Tran.*, 10(5), 635-638, 1970
- 1030 Rudneva, N. M. Position of the Boundary of the Magnetosphere During 1963-1968, *Geomagn. Aeron.*, 13, 96, 1973
- 1040 Russell, C. T. The Solar Wind and Magnetospheric Dynamics, in *Correlated Interplanetary and Magnetospheric Observations*, ed. Page, 3-47, D. Reidel, 1974
- 1050 Russell, C. T. The Control of the Magnetopause by the Interplanetary Magnetic Field, in *Dynamics of the Magnetosphere*, ed. Akasofu, 3-21, D. Reidel, 1979
- 1060 Rycroft, M. J. The Magnetospheric Plasmopause and the Electron Density Trough at the Alouette 1 Orbit, *PSS*, 18, 65, 1970
- 1070 Sandholt, P. E. Effects of Interplanetary Magnetospheric Substorm Variation on the Dayside Aurora, *PSS*, 31, 1345, 1983
- 1080 *Sandholt, P. E. Large- and Small-Scale Dynamics of the Polar Cusp, *JGR*, 90, 4407 1985
- 1090 Schatten, K. H. Response of the Geomagnetic Activity Index Kp to the Interplanetary Magnetic Field, *JGR*, 72, 5185-5191, 1967
- 1100 Schatten, K. H. Large-Scale Properties of the Interplanetary Magnetic Field, *Solar Wind*, NASA Spec. Publ. SP-308, 65, 1972
- 1110 Sergeev, V. A. Quantitative Dependence of the Polar Cap Electric Field on the IMF Bz-Component and Solar Wind Velocity, *PSS*, 29, 205, 1980
- 1120 *Siscoe, G. L. A Quasi-Self-Consistent Axially Symmetric Model for the Growth of a Ring Current Through Earthward Motion from a Pre-Storm Configuration, *PSS*, 27, 285-295, 1979

- 1130 Siscoe, G. L. A Dst Contribution to the Equatorward Shift of the Aurora, PSS, 27, 997-1000, 1979
- 1140 *Siscoe, G. A Theoretical Relation Between Dst and the Solar Wind Merging Electric Field, GRL, 1, 17-19, 1974
- 1150 Siscoe, G. On the Partial Ring Current Contribution to Dst, JGR, 79, 1110, 1974
- 1170 *Snyder, C. W. The Solar Wind Velocity and Its Correlation with Cosmic-Ray Variations and with Solar And Geomagnetic Activity, JGR, 68, 6361, 1963
- 1175 *Sojka, J. J. An Interplanetary Magnetic Field Dependent Model of the Ionospheric Convection Electric Field, JGR, 91, 11281-11290, 1986
- 1177 Stiles, G. S. Ionosonde Observations of the Northern Magnetospheric Cleft During December 1974 and January 1975, JGR, 82, 67, 1977
- 1180 *Su, S.-Y. Magnetic Field Depression at the Earth's Surface Calculated From the Relationship Between the Size of the Magnetosphere and the Dst Values, JGR, 80, 195-199, 1975
- 1190 *Sugiura, M. Quiet Time Magnetospheric Field Depression at 2.3-3.6 Re, JGR, 78, 3182-3185, 1973
- 1200 Svalgaard, L. The Relation Between the Azimuthal Component of the Interplanetary Magnetic Field and the Geomagnetic Field in the Polar Cap, in Correlated Interplanetary and Magnetospheric Observations, ed. Page, 61-84, D. Reidel, 1974
- 1210 Svalgaard, L. Geomagnetic Activity: Dependence on Solar Wind Parameters, in Coronal Holes and High-Speed Wind Streams, ed. Zirker, Colo. Assoc. U. Press, Boulder, Colo., 1977
- 1220 Taylor, H. A. Contraction of the Plasmasphere During Geomagnetically Disturbed Periods, JGR, 73, 961, 1968
- 1230 Troshichev, O. Interplanetary Magnetic Field and Polar Cap Magnetic Distributions: Using the Data for Predictions of Auroral Electrojet Activity, in Solar Terr. Pred. Proc., ed. Donnelly, US Dept. Comm., NOAA/ERL/SEL, Boulder, Colo., 1980
- 1240 *Troshichev, O. The Relationship Between Interplanetary Quantities and Magnetic Activity in the Southern Polar Cap, PSS, 33, 415-419, 1985
- 1250 Tsurutani, B. Interplanetary Magnetic Field Variations and Substorm Activity, JGR, 77, 2964, 1972
- 1260 Wilcox, J. M. Influence of Interplanetary Magnetic Field and Plasma on Geomagnetic Activity During Quiet Sun Conditions, JGR, 72, 19-26, 1967
- 1290 Verzariu, P. Geomagnetic Field Variations Caused by Changes in the Quiet-Time Solar Wind Pressure, PSS, 20, 1909-1914, 1972
- 1300 Yashuhara, F. Equatorward Shifts of Cleft During Magnetospheric Substorms as Observed By ISIS I, JGR, 78, 7286, 1973
- 1310 Yoshizawa, K. Influences of IMF-Sector Polarity on the N-S Asymmetry of Geomagnetic Activity and Its Relationship with Solar Wind Parameters, PSS, 32, 605-610, 1984

		Kp	Dst	AE	ap	Bx	By	Bz(Bs)	Bz*Bz
1	Stand-Off	390 1020 1030	420 1020 1180	8)	940			388	
2	Cusp Latitude	630 790	630 840 850	300 310 630 1300				200 220 300 310 630 690 850 1070 1080 1177 1300	
3	Plasma Sheet	460 710 780							
4	Plasmapause	320 330 700 1060 1220	320						
5	PC Latitudes	340 380 410 420 500 1175	10 1120 1130	380 382 775		80 830	80 830	80 380 382 530 590 830	530
6	PC Potential	530 540 670 700 970 1175		970 1110		427	427	427 970 1110 1175	427
7	PC Precipitation	290 495 502		768 775		975	290 975	290 495 768	
8	Kp	xxxxxxxxxxxx xxxxxxxxxxxxxx					880	170 560	170 1090
9	Dst		xxxxxxxxxxxxxx xxxxxxxxxxxxxx					250 600 610 660 900 920 1040	
10	AE		40	xxxxxxx xxxxxxx		600	800 880	100 130 140 450 490 600 800 820	490
11	AL					220	510 890	510 600 610 900	740
12	AU					220		600 610	740
13	Ap	560			xxxxx xxx			370 490	490
14	Am								740
15	N(S W Density)	980	980					980	980
16	Tail Lobe Ener. D.	1010						260 270 283	520
17	U(magn. ener. dis.)		60					140	
18	Delta B(magn)		1190						

[illegible]

	B Var.	Tilt	SW(T)	SW(deg)	PC Lat	SORT
1					1020	a
2		200				b
3						d
4						e
5	960	420				f
6						g
7			975			h
8	150 560		170			i
9						j
10	490			800		k
11	740 750 890	90 890	750	740 750		l
12	740 750	90	750	740 750		m
13	490 620					n
14	740 750		750	740 750		o
15	980					q
16						r
17						s
18						u